TRANSLATOR GUIDE FOR RCRAINFO VERSION 1 HAZARDOUS WASTE HANDLER IDENTIFICATION & PROGRAM ACTIVITY TRACKING

RCRA_Info_

This page intentionally blank

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1.0   INTRODUCTION

1.1   Background

The purpose of this document is to provide detailed flat file specifications for RCRAInfo System translators. The legacy RCRIS (Resource Conservation & Recovery Information System) and BRS (Biennial Reporting System) national data systems are being migrated from mainframe-based FOCUS applications to a Web-based Oracle application by the Environmental Protection Agency's Office of Solid Waste (OSW).

RCRAInfo consolidates RCRIS and BRS into one national system. A data entry subsystem for the Handler Identification, Permitting, Corrective Action, and Compliance Monitoring and Enforcement subject areas is being developed and will be supported by OSW.   No national data entry subsystem will be maintained as in the past for Biennial Report data.   This function has been privatized.   EPA Regions and States must obtain a Commercial Off-The-Shelf (COTS), Government Off-The-Shelf (GOTS) package, or develop their own software.

States and EPA Regions will continue to have the option to be translators for the RCRA hazardous waste handler identification and program activity tracking component of RCRAInfo; that is, to use their own software and procedures to extract and format data in flat file format and provide it to EPA for transfer to the national system.   Two types of translators will be supported: *full* and *partial*. A *full* translator provides all of their data to EPA in standard, flat file format.   A *partial* translator uses RCRAInfo to enter data for one or more subject areas (modules), and its own system for the rest.

This document is for use by translators who provide hazardous waste data for all or some of the following four subject areas: Handler Identification (HID), Permitting/Closure/Post-Closure (PMT/CL/PC), Corrective Action (CA), and Compliance Monitoring and Enforcement (CM&E). Refer to *RCRAInfo File Specification Guide--1999 Hazardous Waste Report Submissions* for information concerning providing data to EPA for the next RCRA biennial reporting cycle.

1.2   Overview of Document

To be completed at a later date.

1.3   Translator Support

Questions about this document should be directed to the RCRAInfo help desk at rcrainfo.help@epa.gov. Questions submitted to the help desk may only relate to the file specifications for submission of data from the States or EPA Regions which will be loaded into the RCRAInfo National Implementer Database.   The RCRAInfo support staff cannot be responsible for responding to questions about translator system-specific software.

1.4   System Architecture

RCRAInfo is a web-based, Oracle application.   RCRAInfo utilizes a web-browser interface for data entry functions and an Oracle DBMS residing on a central EPA Unix server for data storage. In addition, OSW is using the Cool:Gen Computer Aided Software Engineering (CASE) tool from Sterling Software, Inc. to manage the communications interface between the Oracle database and the web.

Exhibit 1

RCRAInfo System Architecture for HID/Program Activity Data
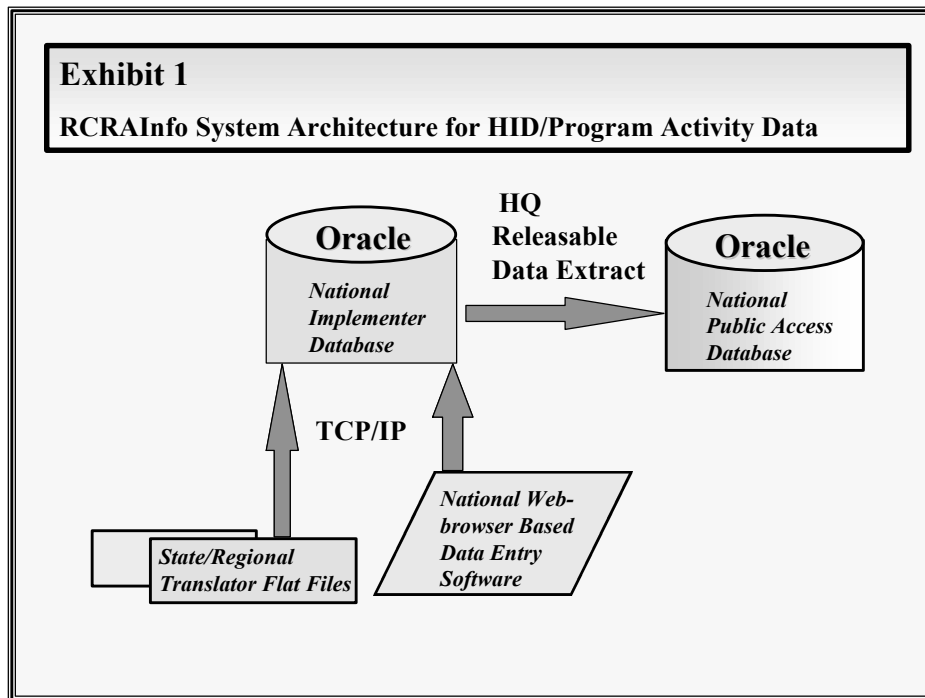
Exhibit 1 - RCRAInfo System Architecture for HID/Program Activity Data depicts a simplified model of the RCRAInfo system architecture for the hazardous waste handler identification and program activity tracking component.   A National Implementer Database will be maintained by the States and EPA Regions. *Implementers* are state and EPA regional staff who implement the RCRA program.   States and Regions *co-implement (co-regulate)* the RCRA program. The National Implementer Database will contain both nationally required and implementer-specific data.   *Nationally required* data (formerly called *core* data) is data needed by EPA Headquarters RCRA staff for program oversight purposes and is expected to be provided as appropriate.. All authorized RCRAInfo users in the states, regions, and at headquarters will have default READ access to all data contained in the National Implementer Database. However, national program oversight reports such as the OSW Program Accomplishments Reports (PARs) will continue to rely only on *nationally required* data. In addition, data extracts for public access purposes will pull only data designated as *nationally required* and non-enforcement sensitive.

A translator will submit flat files in the standard format described in this document.   The data will be assessed and loaded into the database. How often this is done will be negotiated between a Region and State via their annual Memorandum of Agreement (MOA)/Memoradum of Understanding (MOU) process.   Translators will be expected to provide data for loading into the RCRAInfo National Implementer database on at least a monthly basis.

1.5   Relational Database Terminology

Relational database terminology appears throughout this document. This section provides a crosswalk between relational concepts and terms utilized by previous generations of database software packages. The standard flat files to be produced by translators closely mirror the Oracle table structure of RCRAInfo.

Oracle is a relational database management system (DBMS). In a relational DBMS, data is stored in *tables*. A table is composed of *columns* and *rows*. Columns are the same as *data elements, fields* or *attributes*. Rows are the same as *records*. One or more columns that together uniquely identify a row are called the *key* to the table. Tables can be related to each other by a column or group of columns that contain common information. *Referential Integrity* defines a correct relationship between columns and tables in a relational database. When two tables contain common information in a column or set of columns, and those common column(s) form the unique key in one table, then the latter table is called a *parent table* and its key is called a *parent key*. The other table is called a *child table*, and the relation between the tables is called a *parent-child* relationship. The child table is said to *depend* on the parent table. The common information in the child table is called a *dependent* or *foreign key*. When a parent record is deleted, referential integrity requires the deletion of all child records with a foreign key value which matches the deleted parent key value. This is called a *cascade delete*.

The primary, parent key for the Handler Identification and Program Activity Tracking tables in RCRAInfo is the Handler ID. The highest parent table is the Basic Handler (HBASIC) table. In RCRAInfo, all tables are related to one another using the Handler ID number, which is used in all *subject areas* to tie information together. Another term for subject area is module. An example of a RCRAInfo subject area is Handler Identification. A child table of HBASIC is the Permit Series Table (PSERIES). Handler ID is the parent key in HBASIC and the foreign key in PSERIES.

1.6   Other Documents/Files

- *RCRAInfo Data Element Dictionary*
- *RCRAInfo Table Lists*
- *RCRAInfo Entity Relationship Diagrams (ERDs)*
- *RCRAInfo Lookup Table Files*
- *RCRIS to RCRAInfo Data Crosswalks*

The above supplemental information will be posted in PDF format as it becomes available on the Office of Solid Waste Partnership Web Site at http://www.epa.gov/oswfiles.

2.0   DATA SUBMISSION OVERVIEW

2.1   Testing the Translation Effort

It is the responsibility of the state or regional translator to ensure that correctly formatted flat files are produced for inclusion in the RCRAInfo National Implementer Database.

2.2   Data Transfer Requirements

Translator flat files must ultimately be transmitted to the Unix server located at EPA's National Computer Center in North Carolina where the RCRAInfo National Implementer Oracle Database resides. Responsibility for this transmittal and for loading the data into the National Implementer database will be agreed upon by the State and the appropriate EPA Regional RCRA personnel.   The preferred method for providing data to EPA is electronic file transfer via Internet-based File Transfer Protocol (FTP).   For large submissions, file compression is recommended. Other options are mailing floppy diskettes, sending data in an attachment to an EMAIL message, or some other form of electronic data transfer. Submission of data using magnetic tape cannot be supported. State translators who utilize any method other than FTP for file transfer must submit their data to the EPA Regional Database Administrator who will then FTP the data to the EPA Unix server.

Before sending data, the following steps are recommended in order to ensure that the data transfer process proceeds smoothly:

● The transfer medium should be tested after completion of the data extract process to ensure readability and correctness of the data.
● If a physical medium (e.g., diskette) is used, the medium must be completely and accurately labeled. Any instructions required to correctly retrieve that flat files from the medium should also be included. In addition, the data submission should be sent in packaging which will protect it from magnetic and/or static electric disturbance.
● A backup copy of the data files should be retained as a record of the submission, and for use in case the original submission is lost or damaged.

2.3   Control Information

The State or Regional translator must produce a Control File (flat file ID = 'CL' as described in Appendix A) to be included with each data submission which summarizes each flat file being submitted. One Control File record is created for each submitted flat file. **All data in the Control File must be in upper case**. The following fields must be included in the Control File:

● Flat file Name (see Appendix A)
● Version of RCRAInfo data entry software currently in production (first digit only)
● Creation date for the flat file
● Number of records in the Flat File
● Module full replacement indicator which indicates whether all records for a subject area (module) are being supplied.

Translator load jobs will be submitted for one state at a time. Ownership of the data being submitted will be derived from information provided via the job submittal data entry process. Translators will not need to provide this information via the flat files.

It is recommended that the translator develop a standard set of audit reports at a greater level of detail than provided in the control file for use in troubleshooting should problems occur during loading of the data into RCRAInfo.

2.4   Data Ownership/Security

Implementer of Record (IOR) is the term used to describe the agency (State or EPA) responsible for data entry and management of specific RCRAInfo handler identification and program activity data for a given state or territory. Usually, the responsible agency that does the work owns the data that tracks the work and, therefore, is the IOR for that data. A State and EPA are responsible for assigning the IOR for RCRA data during annual RCRA grant negotiations. The annual RCRAInfo data responsibilities become a component of the State/EPA Memorandum of Agreement (MOA) process where applicable, and a Memorandum of Understanding (MOU) for information sharing and data management responsibilities between the State and EPA.

Although all implementer data in RCRAInfo will reside in the same physical database, each implementer will own and control update access to its data. Security functions implemented at both the database- and application levels will prevent unauthorized users from changing data. Please note that all **authorized** RCRAInfo application users will be allowed universal read access to all data in the database, including implementer-defined information and enforcement sensitive information.

There are two types of IOR present in RCRAInfo: *Fixed* and *Variable*. Fixed IOR applies to data tables for which either the state or the region is completely responsible. Variable IOR applies to data tables where ownership may vary between the state and the region depending on specified values in a decision column/field.

IOR for fixed IOR tables is determined by examining the first two characters of the Handler ID for which data is being entered. For example, in the PSERIES table, the first two characters of FK_HBASICHANDLER_ID are utilized to determine row ownership. **This means that ALL state ID numbers where used as the primary key must begin with a valid state prefix**.

IOR for variable IOR tables is determined by checking a data column called ACTIVITY_LOCATION and the decision column for the table. For example, in the PEVENT table ACTIVITY_LOCATION and RESPONSIBLE_AGENCY are queried. Allowed values for ACTIVITY_LOCATION are all state postal codes. Allowed values for RESPONSIBLE_AGENCY are E, S, or J. A table (called TSRESPONSIBLE_PARTY) is maintained and queried in RCRAInfo which tracks which agency owns each value for each decision column (in this instance, E, S, and J).

RCRAInfo introduces a new data tracking capability which impacts the way IOR is implemented in the system. Regions and States may enter data into tables where variable IOR applies, and link that data to data in fixed IOR tables associated with Handler ID Numbers owned by other States or Regions. The primary purpose of this change is to resolve the long-standing out-of-state transporter ID number assignment and tracking issue. Regions and States may now enter their own Handler Identification, Permitting, Corrective Action, and Compliance Monitoring & Enforcement data and link it to an out-of-state ID number. Each Region/State co-implementer pair will continue to negotiate which implementer owns fixed IOR data. The difference now is that IOR assignments for co-implementer pairs in other Regions and States will also determine which one will be allowed to link variable IOR data to fixed IOR data for out-of-state ID numbers.

The exception to the rule for fixed IOR tables is the HHANDLER table. HHANDLER is a fixed IOR table where the data column ACTIVITY_LOCATION will be used to determine which co-implementer within a Region/State pair can create and maintain Handler Identification subject area data for an out-of-state HBASIC Handler ID number.

The *RCRAInfo Translator Flat File Load Utility* software will enforce IOR security settings that are maintained against the RCRAInfo National Implementer Oracle database by each Regional Database Administrator.   Data not owned by the implementer submitting the translator files will not be loaded. Ownership will be derived from the User ID of the implementer submitting the translator load. and will be used to match each data row against the database IOR settings.   The Flat File Specifications provide the IOR type (variable or fixed) and the IOR decision column, where applicable, for each flat file.

2.5   Quality Assurance

In addition to enforcing Implementer of Record (IOR) rules, the *RCRAInfo Translator Flat File Load Utility* software will validate incoming translator data.   Refer to **Appendix A – Translator Flat File Specifications** and **Appendix B – Translator Data Assessment Edits** for details.

Eight categories of edit checks will be performed, depending upon the specific column (field) or related columns being edited:

- Control File Checks
- Referential Integrity (Orphan Checks)
- Mandatory/System Required Column Checks
- Table Lookups
- Hard-coded Value Checks
- Cross Reference Checks
- Data Type Checks
- Relational Operator Checks

2.5.1   Control File Checks

The first step in the translator data load process will be to perform *pre-load* checks to ensure that the information provided in the control file matches what is contained in the other submitted flat files (See **Appendix A–Flat File Specifications).** For example, if the actual number of records in a particular flat file does not match number of records field contained in the control file, an error flag will be set.   Any control file errors that are detected will result in the termination of the translator data load process.

2.5.2   Referential Integrity (Orphan Checks)

The *RCRAInfo Translator Flat File Load Utility* will verify that the parent and foreign keys provided via the flat files match up in the database. This is critical to ensuring that no orphans are created in the database (child rows with no parent row). For example, the most basic referential integrity check is that a Handler ID number must exist in the HBASIC table (H0 flat file) in order to create an associated row in another table such as HHANDLER (H1 flat file) or PSERIES (P1 flat file).

2.5.3 Nationally Required/System Required Column Checks

A *nationally required* data column is considered necessary for program oversight purposes. A *conditional nationally required* data column is necessary if certain conditions are true. Translators must provide nationally required data as applicable. The responsibility to provide non-nationally required data is negotiated between the Region and the State via the State/EPA Memorandum of Agreement (MOA) process. Information sharing and data management responsibilities between the State and EPA are documented in a Memorandum of Understanding (MOU).

System required data columns are necessary for data integrity and for proper system functioning. Like certain nationally required data columns, some system-required data columns are required only if certain conditions occur. Refer to **Appendix A–Flat File Specifications** for detailed information on nationally required and system required data columns.

2.5.4 Table Lookup Checks

Where there are codes (valid values) associated with a particular column (field) in the database, a lookup table is most often maintained in RCRAInfo. Examples of lookup tables in RCRAInfo are the County Code Table (LU_COUNTY) and the Permit Event Code Table (LU_PERMIT_EVENT). RCRAInfo data entry software employs pick-lists whereby users highlight and select a valid value from the table. The codes are stored in the database as foreign keys to the lookup table. Translators who provide implementer-defined codes to EPA must ensure that these codes are added to the appropriate RCRAInfo lookup table. Most RCRAInfo lookup tables allow implementer-defined codes to be added and maintained.

2.5.4.1 Code Ownership

The column OWNER found in each RCRAInfo lookup table plays an important role in controlling what codes are displayed on a pick list during the data entry process. The allowed values for OWNER are *HQ*, *US*, *01 - 10*, or a valid postal state prefix. HQ (or EPA Headquarters) codes are codes that are nationally defined and nationally required (formerly called *core*). US (for United States) codes are codes that are nationally defined, but are not nationally required (formerly called *non-core*). Codes 01 - 10 represent the ten EPA Regions, and the postal state prefix codes, of course, represent the States. Codes owned by Regions and States are implementer defined and not nationally required. RCRAInfo users and translators will be allowed to add, update, and delete codes from the database only where the associated code OWNER = HQ **or** US **or** the Region/State group of which the user (or translator) is a member.

> **Note**: **Translators must supply the OWNER for any column which is associated with a lookup table as indicated in *Appendix A - Flat File Specifications*.**

2.5.4.2 Active Status

The column ACTIVE_STATUS found in each RCRAInfo lookup table controls which codes can be added to the database via the RCRAInfo data entry screens. Only codes with an active status = *Y* will appear on pick lists. Codes with an active status = *N* can remain in the database as historical codes. Likewise, translators will be allowed to create rows containing inactive codes when the full replacement option is chosen. However, if transaction processing is used, an inactive code will be rejected if the translator attempts to add one. (See *Section 3.6-Flat File Processing Methodology* for discussion of full-replacement and transaction processing options.)

2.5.5   Hard-coded Value Checks

In the few instances where lookup tables are not used, the RCRAInfo Translator Data Assessment software will be hardcoded with the valid values for a particular column (field) being validated. In all cases, this type of check is used when there are only a few values to be checked. One example is the Responsible Agency (RESPONSIBLE_AGENCY) column in the Permitting Event (PEVENT) table. Responsible Agency must be equal to **E**, **S**, or **J**. A second example is the Transaction Code (TRANS_CODE) which translators must include in each flat file in order to enable transaction processing. Valid values for TRANS_CODE are **C** or **D** to indicate transaction type or **blank** if the Full Replacement Indicator (FULL_REPLACE_IND) in the translator Control File is equal to **Y** for the flat files in the module.

2.5.6   Cross Reference Checks

The last example provided in the previous section also illustrates the fact that multiple related columns (fields) often must be validated together because there are dependencies between them. A common type of cross reference check is where a particular column cannot be blank if there is a value in another related column. An example of this is that for a row in the Owner/Operator (HOWNER_OPERATOR) Table, Previous Owner/Operator Date (OWNER_OPERATOR_CHANGE_DATE) is required if the Owner/Operator Indicator (OWNER_OPERATOR_INDICATOR) = **PO** (previous owner) or **PP** (previous operator).   Referential integrity checks and certain conditional nationally required checks are also examples of cross reference checks.

2.5.7   Data Type Checks

For RCRAInfo data, a data type check is used only for columns defined as numeric. Examples are CAPACITY and NUMBER_OF_UNITS in the Permitting Unit Detail (PUNIT_DETAIL) table. Only numeric digits are allowed in these columns. Numeric columns are defined as integer, decimal or date format.

2.5.8   Relational Operator Checks

The term relational operator check is being used here to refer to numeric columns which must be greater than, less than, or between a range of a specified value(s). Examples are schedule date and actual date for a Permitting event. Schedule dates (SCHEDULE_DATE_ORIG and SCHEDULE_DATE_NEW) must be a valid date greater than 5/18/1980. ACTUAL_DATE cannot be greater than the current date.

2.6   Flat File Processing Methodology

Because the RCRA program is co-implemented (i.e., data from both the Region and the State may be provided for a particular Handler ID), RCRAInfo supports transaction processing for translator flat files. Implementer of Record rules are enforced to ensure that the translator does not change or delete data owned by the co-implementer. Transactions will be processed in sequential order as they are read from the flat file. This means that if a transaction to delete data occurs, the data may be added back with a subsequent change transaction (if enough data is provided so as not to generate an error(s)).

A Full Replacement option by subject area (module) is also available. **Please note that the Full Replace option will *only* be allowed if the translator owns all of the data in the module and FULL_REPLACE_IND in the Control File records for the flat files for the module is set to Y.** A field called Transaction Code (TRANS_CODE) is used for all flat files except the Control File. The allowed values for TRANS_CODE are:

C	Change (Add or Update)
D	Delete
Blank	Full Replace

For *Change* transactions, data will be added if parent key values do not currently exist in the database, and will be modified if parent key values exist in the database. Full replacements for the row being changed must be provided.

For *Delete* transactions, parent key values must currently exist in the database, and only child rows owned by the translator can be present. Only the transaction code and the key fields necessary to identify the parent row to be deleted need to be provided. In addition, a check will be performed as appropriate prior to deletion of a row to ensure that no data owned by another implementer is linked to that row or any other row in related tables impacted by a *cascade delete*.

2.7 Handler Identification Subject Area

The objective of the Handler Identification Subject Area is to maintain an inventory of hazardous waste handlers to support RCRA program activity tracking and public data access requests. Hazardous waste handlers are required to notify EPA of their existence by submitting identifiying information on the Federal Notification of Regulated Waste Activity Form (EPA form 8700-12) or a State equivalent form. This form provides the States and EPA with basic handler identification data concerning what and where (name, address, owner), specific hazardous waste activities, and wastes handled. EPA or State inspections provide confirmation or updates to information submitted by regulated entities, or may identify entities (called non-notifiers) who are actively generating, transporting, treating, storing, or disposing of hazardous waste, but have not notified EPA.

In RCRAInfo, the parent table for all handler identification and program activity tracking data is the HBASIC table. This provides the entry point for adding new handlers into the system. A row for a handler must exist in the HBASIC table before data in any other table may be added or updated. Each handler must be assigned a unique *Handler Identification Number* which serves as the primary key for the entire system, and a *Handler Name* must be provided. It is also mandatory to indicate via the *Extract Flag* whether nationally required and non-enforcement sensitive data may be extracted for this handler for public access purposes. A place-holder for the EPA Facility Registry System (FRS) *Facility Identifier* is maintained here. The HBASIC table contains one row per ID number, and, therefore, one *owner* per row. A one-to-many relationship between the HBASIC and HHANDLER tables supports *Out-of-State Transporter* tracking. Implementers may use the actual ID number for entering data for out-of-state transporters instead of assigning a different one with their own state code prefix. HHANDLER may contain multiple rows per ID number, each owned by a different implementer (and having a different activity location).

Additional handler identification data is stored in the HHANDLER table. *Location-* and *Mailing Address* information is included. However, *Street Number* may be optionally parsed from *Street* for both address types to facilitate searching and sorting. Handler-level indicators such as *Type of Land*, *Non-Notifier*, *Facility Existence Date*, and *Handler Accessibility* are stored in this table. Unlike the HBASIC table, the

HHANDLER table may contain more than one row for a single Handler ID number. The *Latitude* and *Longitude* coordinates for a facility are stored in the HLATITUDE_LONGITUDE table. Only a single set of coordinates may be provided. Implementers may optionally provide Environmental Priority Ranking history for a handler. The *EPR Status* for a particular date is contained in the HENV_PRIORITY_RANK table.

> **Note:** Street Number length for location and mailing address is currently 6 characters in the translator flat file specifications, but has been lengthened in the RCRAInfo database to 12 characters. Translator software and documentation will be corrected in the next structural revision of the guide.

RCRAInfo introduces the capability to track handler name history. Previous handler names and associated dates are contained in the HPREVIOUS_NAME table. Names, addresses, and phone numbers for Handler Contacts for various contact types (such as for Notification and Part A forms) are contained in the HCONTACT table. The HOWNER_OPERATOR table stores information about owners and operators for a handler such as name, type, address, and phone number. Multiple owner and operator rows may be supplied and a history of previous owners and operators may also be maintained. For previous owners and operators, a date changed must be provided.

Standard Industrial Classification (SIC) codes for a facility are tracked in the HSIC table. One and only one SIC code must be designated as primary. The HOTHER_PERMIT table stores information on other environmental permits provided by the facility on the federal Part A permit application form. This data is optional.

> **Note:** The capability to track NAICS codes will not be added to RCRAInfo until the first structural revision if, at that time, policies and procedures have been promulgated by the agency.

In RCRAInfo, implementers have the option to track multiple views of handler activities and waste stream data. A SOURCE_TYPE of **N** indicates that the information was reported by the handler on the federal Notification form or state equivalent. A SOURCE_TYPE of **A** indicates that the information was reported by a facility on the federal Part A permit application form or state equivalent. A SOURCE_TYPE of **E** indicates EPA inspection; a SOURCE_TYPE of **S** indicates State inspection. If only one view is tracked, a SOURCE_TYPE of **N** is used.

For each SOURCE_TYPE, at indication of at least one handler activity (such as Generator or Transporter) must be provided unless the handler is a non-notifier. The HACTIVITY table tracks information on handler activity and federal and state regulatory status. Waste code data which is collected on the federal Notification and Part A forms is optional. The HHAZARDOUS_WASTE_STREAM, HHAZARDOUS_WASTE_CODE, and HHAZARDOUS_WASTE_PROCESS tables contain this information.

2.7.1 Handler Identification Subject Area Entity Relationships

Exhibit 2A
Handler Identification
Subject Area Entity
Relationship Diagram
PART 1
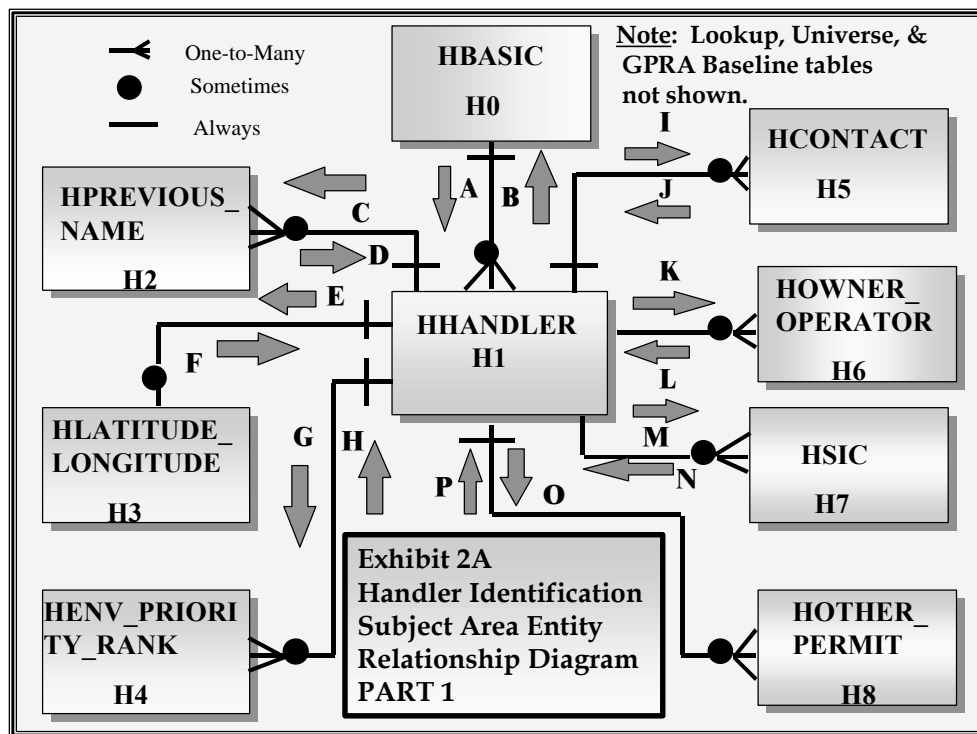
Exhibit 2A provides a graphical depiction of the first nine tables in the Handler Identification Subject Area, and the relationships between them. Refer to the legend below for a description of each relationship. Each table is directly related to a corresponding flat file. The "crows feet" indicate a one-to-many relationship between rows in two linked tables. Thel bars indicate mandatory relationships (ie., linkages are required in all cases). The circles indicate optional relationships (i.e., linkages are not required). Lookup tables and Universe tables (which contain data calculated by the RCRAInfo system), and the GPRA Baseline table are not shown because translators will not provide them as flat files.

A. Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** Handler (HHANDLER) rows.

B. Each Handler (HHANDLER) row **must always** be linked to **one** Handler Basic (HBASIC) row.

C. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Previous Name (HPREVIOUS_NAME) rows.

D. Each Previous Name (HPREVIOUS_NAME) **must always** be linked to **one** Handler (HHANDLER) row.

E. Each Handler (HHANDLER) row **may or may not** be linked to **one** Lat/Long (HLATITUDE_LONGITUDE) row.

F. Each Latitude/Longitude (HLATITUDE_LONGITUDE) row **must always** be linked to **one** Handler (HHANDLER) row.

G. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Environmental Priority Ranking (HEP_RANKING) rows.

H. Each Environmental Priority Ranking (HENV_PRIORITY_RANK) row **must always** be linked to **one** Handler (HHANDLER) row.

I. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Contact (HCONTACT) rows.

J. Each Contact (HCONTACT) row **must always** be linked to **one** Handler (HHANDLER) row.

K. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Owner/Operator (HOWNER_OPERATOR) rows.

L. Each Owner/Operator (HOWNER_OPERATOR) row **must always** be linked to **one** Handler (HHANDLER) row.

M. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Standard Industrial Classification Code (HSIC) rows.

N. Each Standard Industrial Classification Code (HSIC) row **must always** be linked to **one** Handler (HHANDLER) row.

O. Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Other Permits (HOTHER_PERMIT) rows.

P. Each Other Permit (HOTHER_PERMIT) row **must always** be linked to **one** Handler (HHANDLER) row.

Exhibit 2B

Handler Identification
Subject Area Entity
Relationship Diagram

PART 2

One-to-Many

Sometimes

Always

HHANDLER
H1

A   B

C   HSOURCE
HA   E

D   F

HACTIVITY
HB

G   HHAZARDOU
S_WASTE_ST
REAM   HC   I

H   J

HHAZARDOU
S_WASTE_CO
DE   HD

HHAZARDOUS
_WASTE_PRO
CESS   HE

Note: Lookup
tables not shown
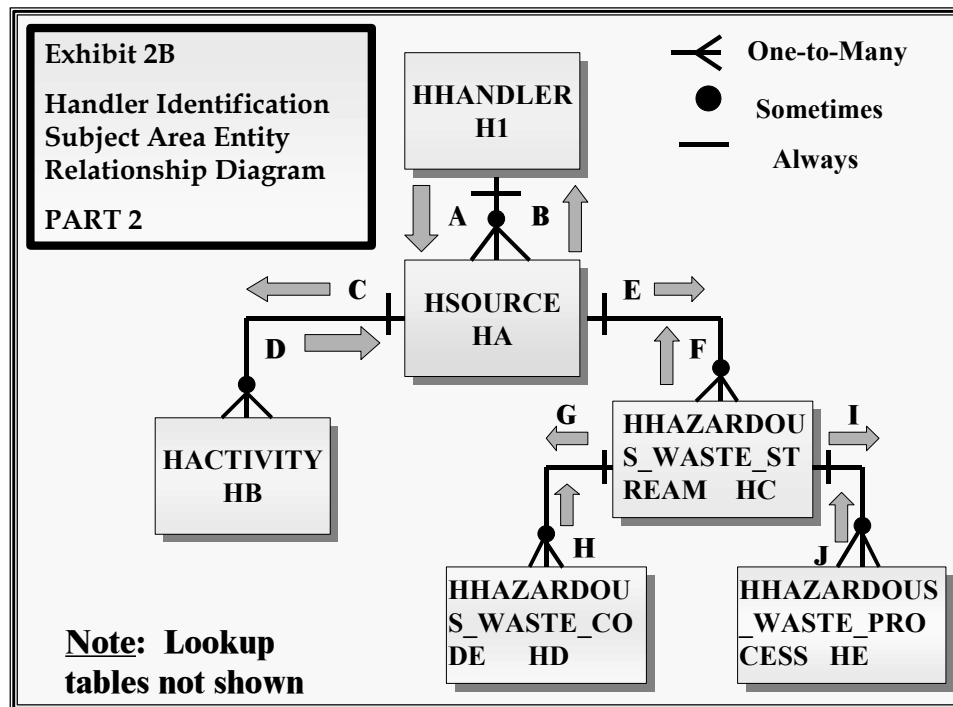
Exhibit 2B provides a graphical depiction of the remaining tables in the Handler Identification Subject Area, and the relationships between them.   Refer to the legend below for a description of each relationship. Each table is directly related to a corresponding flat file. The "crows feet" indicate a one-to-many relationship between rows in two linked tables. The bars indicate mandatory realtionships (i.e., linkages are required in all cases). The circles indicate optional relationships (i.e., linkages are not required). Lookup tables are not shown because translators will not provide them as flat files.

A.   Each Handler (HHANDLER) row **may or may not** be linked to **one or more** Source (HSOURCE) rows.

B.   Each Info. Source (HSOURCE) row **must always** be linked to **one** Handler (HHANDLER) row.

C.   Each Info. Source (HSOURCE) row **may or may not** be linked to **one or more** Handler Activity (HACTIVITY) rows.

D.   Each Activity (HACTIVITY) row **must always** be linked to **one** Info. Source (HSOURCE) row.

E.   Each Info. Source (HSOURCE) row **may or may not** be linked to **one or more** Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) rows.

F.   Each Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) **must always** be linked to **one** Info. Soruce (HSOURCE) row.

G.   Each Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) row **may or may not** be linked to **one or more** Hazardous Waste Code (HHAZARDOUS_WASTE_CODE) rows.

H.   Each Hazardous Waste Code (HHAZARDOUS_WASTE_CODE) row **must always** be linked to **one** Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) row.

I.   Each Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) row **may or may not** be linked to **one or more** Hazardous Waste Associated Process (HHAZARDOUS_WASTE_PROCESS) rows.

J.   Each Hazardous Waste Associated Process (HHAZARDOUS_WASTE_PROCESS) row **must always** be linked to **one** Hazardous Waste Stream (HHAZARDOUS_WASTE_STREAM) row.

2.7.2   Handler Identification Transaction Processing

A transaction code (TRANS_CODE) must be present in each flat file for each record to indicate whether the modification is an *Add/Update* (TRANS_CODE = **C**) or a Delete (TRANS_CODE = **D**), **and** the Full Replacement Indicator (FULL_REPLACE_IND) in the Control File for all submitted flat files associated with the module must = **Blank**.

2.7.2.1   H0 (HBASIC) Flat File

HBASIC data is part of the fixed IOR HHANDLER (H1) block; either the State or Region owns all of the data in this table for a given state or territory (activity location). Only one HBASIC row may exist in the database for a given Handler Identification Number. The translator must be IOR for HHANDLER data for a given ACTIVITY_LOCATION to effect any changes to the table.

2.7.2.1.1   Add/Update

To add or update a Handler Basic table row, TRANS_CODE must = **C**.

●   To update a HBASIC table row, a replacement for the entire row must be provided. Column-specific updates are not suppported. If a row owned by the translator exists in HBASIC which matches the key column HANDLER_ID provided in the H0 flat file, the row will be updated.

> **Note: A Handler Identification Number cannot be changed using a *Change* transaction because it is the primary key for the system. To change a handler identification number, all rows in the database for the handler must first be deleted and then recreated (added) using the revised key.**

●   If a HBASIC table row which matches the key field HANDLER_ID provided in the H0 flat file does **not** exist **and** if the translator is authorized, a new HBASIC table row will be created.

> **Note: A HBASIC row should be linked to one or more HHANDLER table rows even though the database structure doesn't enforce it. When adding a new handler to the database, translators should also include corresponding handler identification data in the H1 (HHANDLER) flat file.**

Note: Implementers may use non-standard Handler ID numbers as the primary key. However, the first two characters of the non-standard ID number <u>must</u> be a valid state code. If a non-standard Handler ID number is added to HBASIC, the Secondary ID Number field in HHANDLER (H1) must be populated with the same ID number. The translator load data asssessment software will apply the EPA ID number check digit algorithm to all HANDLER_IDs added to HBASIC.   If the ID number provided in H0 fails the check digit algorithm, a WARNING message will be displayed and the handler will be added to the database. When the H1 (HHANDLER) flat file is subsequently processed, the data assessment software will check to see if a secondary ID number (SECOND_ID) is provided which matches the non-standard ID number added to the HBASIC table earlier in the process. If not, the H1 record will be rejected.

2.7.2.1.2   Delete

Caution must be exercised with this transaction. Deleting a row from the HBASIC table removes the HBASIC table row PLUS <u>all</u> rows in other tables associated with the specified Handler Identification Number. In order to delete a handler from the database, the H0 flat file must contain a record where TRANS_CODE = **D** and the key field HANDLER_ID is provided. A matching row owned by the translator must exist in the database.

Rows in other flat files representing the delete Handler transaction must **not** be present. Deleting a Handler will automatically *cascade delete* all rows in other related RCRAInfo Handler Identification and Program Activity Tracking tables.   Related delete transactions in other flat files will generate error messages because the data will have been previously deleted via the H0 flat file "Delete Handler" transaction.

> **<u>Note:</u> If any entries exist in any other tables for a given Handler ID which have an owner other than the owner specified in the translator control file, the transaction will not be processed. Before deletion of the handler can proceed, the co-implementer and/or other implementers must delete all related data (i.e., any rows linked to the specified Handler Identification Number) from the RCRAInfo National Implementer database.**

2.7.2.2   H1 (HHANDLER) Flat File

HHANDLER data is fixed IOR within the context of a Region/State co-implementer pair; either the State or the Region owns all of the data in this table for a given state or territory. Only one HHANDLER row per Handler ID may exist for a particular Region/State co-implementer pair (i.e., ACTIVITY_LOCATION); however multiple HHANDLER rows may exist which are owned by other States or Regions in order to support tracking of out-of-state transporters. In either case, the translator must be IOR for HHANDLER data for the specified ACTIVITY_LOCATION to effect any changes to the table.

2.7.2.2.1   Add/Update

To add or update a Handler Identification data table row, TRANS_CODE must = **C**.

- To update a HHANDLER table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a row owned by the translator exists in HHANDLER which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION provided in the H1 flat file, the row will be updated.

2.7.2.2.1   Add/Update (continued)

- If a HHANDLER table row which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION   provided in the H1 flat file does **not** exist **and** if the translator is authorized, a new HHANDLER table row will be created.

    > **Note: A HHANDLER row must be linked to a matching HBASIC row.   When adding a new Handler Identification data row, the translator must first create the associated HBASIC (H0) row if that translator is the owner of HBASIC for a given state or territory. Regardless, if there is no matching HBASIC row in the database, a referential integrity violation will occur.**

2.7.2.2.2   Delete

If the translator owns HBASIC data for a given state or territory, a delete transaction for HHANDLER data will not be allowed because a referential integrity violation would occur. The translator must instead use a delete transaction in the HBASIC (H0) flat file to remove a handler from the database.

However, translators who do not own the associated HBASIC row and have created Handler Identification data to track an out-of-state transporter **may** delete HHANDLER data. In order to delete a Handler Identification data row, the H1 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID and ACTIVITY_LOCATION are present. Rows in other related flat files representing the delete transaction must **not** be present. Deleting Handler Identification data for a Handler ID will automatically *cascade delete* all associated rows owned by the translator in the other Handler Identification subject area tables (flat files H2 - HE). Related delete transactions in the H2 - HE flat files will generate error messages because the data will have been previously deleted via the H1 flat file "Delete Handler Identification Data" transaction.

2.7.2.3   H2 (HPREVIOUS_NAME) Flat File

Handler Previous Name data is fixed IOR within the context of a Region/State co-implementer pair; either the State or Region owns all of the data in this table for a given state or territory (i.e., ACTIVITY_LOCATION). Each Handler ID for a given ACTIVITY_LOCATION may be associated with multiple previous names.   The translator must be IOR for HHANDLER data to effect any changes to the HPREVIOUS_NAME table.

2.7.2.3.1 Add/Update

To add or update a Handler Previous Name table row, TRANS_CODE must = **C**.

2.7.2.3.1 Add/Update (continued)

●   To update a HPREVIOUS_NAME table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HPREVIOUS_NAME table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** RECEIVE_DATE provided in the H2 flat file, the HPREVIOUS_NAME row will be updated.

●   If a HPREVIOUS_NAME table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** RECEIVE_DATE provided in the H2 flat file does **not** exist, **and** if the translator is authorized, a new HPREVIOUS_NAME row will be created.

> **Note: The date received (RECEIVE_DATE) for a name change cannot be updated via a change transaction because it is a key column. To update an erroneous receipt date, the row must first be deleted and then recreated with the revised key.**

2.7.2.3.2 Delete

To delete a Previous Name row, the H2 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and RECEIVE_DATE are provided. A matching row owned by the translator must exist in the HPREVIOUS_NAME table.

2.7.2.4   H3 (HLATITUDE_LONGITUDE) Flat File

Locational data is fixed IOR within the context of a Region/State co-implementer pair; either the State or Region owns all of the data in this table for a given state or territory. One locational coordinate row per Handler may be supplied for a particular Region/State co-implementer pair (i.e., ACTIVITY_LOCATION). The translator must be IOR for HHANDLER data to effect any changes to the HLATITUDE_LONGITUDE table.

2.7.2.4.1 Add/Update

To add or update a Latitude/Longitude table row, TRANS_CODE must = **C**.

●   To update a HLATITUDE_LONGITUDE table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HLATITUDE_LONGITUDE table row owned by the translator exists which matches the key fields HANDLER_ID and ACTIVITY_LOCATION provided in the H3 flat file, the row will be updated.

●   If a HLATITUDE_LONGITUDE table row owned by the translator which matches the key fields HANDLER_ID and ACTIVITY_LOCATION provided in the H3 flat file does **not** exist **and** the translator is authorized, a new HLATITUDE_LONGITUDE row will be created.

2.7.2.4.2 Delete

To delete a locational coordinate row, the H3 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID and ACTIVITY_LOCATION are provided. A matching row owned by the translator must exist in the HLATITUDE_LONGITUDE table.

2.7.2.5 H4 (HENV_PRIORITY_RANK) Flat File

Environmental Priority Ranking data is fixed IOR within the context of a Region/State co-implementer pair (i.e., ACTIVITY_LOCATION); either the State or Region owns all of the data in this table for a given state or territory. Multiple Environmental Priority rankings per IOR owner may be supplied.   The translator must be IOR for HHANDLER data to effect any changes to the table.

2.7.2.5.1 Add/Update

To add or update an Environmental Priority Ranking table row, TRANS_CODE must = **C**.

● To update a HENV_PRIORITY_RANK table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HENV_PRIORITY_RANK table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EPR_DATE provided in the H4 flat file, the row will be updated.

● If an HENV_PRIORITY_RANK table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EPR_DATE provided in the H4 flat file does **not** exist **and** the translator is authorized, a new HEP_RANKING row will be created.

> **Note: The EPR Date (EPR_DATE) for an EP Ranking cannot be updated via a change transaction because it is a key column. To update an erroneous EPR date, the row must first be deleted and then recreated with the revised key.**

2.7.2.5.2 Delete

To delete an Environmental Priority Rank table row, the H4 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and EPR_DATE are provided. A matching row owned by the translator must exist in the HENV_PRIORITY_RANK table in order to be deleted from the system.

2.7.2.6 H5 (HCONTACT) Flat File

HCONTACT data is variable IOR; the decision field is CONTACT_TYPE.   The translator must be IOR for the specified contact type for the given ACTIVITY_LOCATION in order to create or update contact data.

2.7.2.6.1 Add/Update

To add or update a Handler Contact table row, TRANS_CODE must = **C**.

- To update a HCONTACT table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HCONTACT table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** OWNER (for Contact Type Code) **plus** CONTACT_TYPE provided in the H5 flat file, the HCONTACT row will be updated.

- If a HCONTACT table row owned by the translator which matches the key fields HANDLER_ID **plus** OWNER (for Contact Type Code) **plus** ACTIVITY_LOCATION **plus** CONTACT_TYPE provided in the H5 flat file does **not** exist, a new HCONTACT row will be created.

2.7.2.6.2 Delete

To delete a Handler Contact table row, the H5 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, OWNER (for Contact Type Code), and CONTACT_TYPE are provided. A matching row owned by the translator must exist in the HCONTACT table.

2.7.2.7 H6 (HOWNER_OPERATOR) Flat File

Handler Owner/Operator data is fixed IOR within the context of a Region/State co-implementer pair (i.e., ACTIVITY_LOCATION); either the State or Region owns all of the data in this table for a given state or territory. Each Handler ID may be associated with multiple current and previous owner and operator names. The translator must be IOR for HOWNER_OPERATOR data to effect any changes to the table.

2.7.2.7.1 Add/Update

To add or update a Owner or Operator table row, TRANS_CODE must = **C**.

- To update a HOWNER_OPERATOR table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HOWNER_OPERATOR table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** OWNER_OPERATOR_SEQUENCE provided in the H6 flat file, the HOWNER_OPERATOR row will be updated.

- If a HOWNER_OPERATOR table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** OWNER_OPERATOR_SEQUENCE provided in the H6 flat file does **not** exist, a new HOWNER_OPERATOR row will be created.

> **Note: The Owner/Operator Indicator (OWNER_OPERATOR_INDICATOR), which is used to determine whether the row represents a current owner, previous owner, current operator, or previous operator, is NOT a key column. OWNER_OPERATOR_SEQUENCE, therefore, does not repeat by OWNER_OPERATOR_INDICATOR as may be expected. It must be unique for the HHANDLER row it is linked to.**

2.7.2.7.2   Delete

To delete an Owner/Operator row, the H6 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and OWNER_OPERATOR_SEQUENCE are provided. A matching row owned by the translator must exist in the HOWNER_OPERATOR table.

2.7.2.8   H7 (HSIC) Flat File

Handler SIC Code data is fixed IOR within the context of a Region/State co-implementer pair; either the State or Region owns all of the data in this table for a given state or territory (i.e., ACTIVITY_LOCATION). Each Handler ID may be associated with multiple SIC Codes.   The translator must be IOR for HSIC data to effect any changes to the table.

2.7.2.8.1   Add/Update

To add or update a SIC Code table row, TRANS_CODE must = **C**.

- To update a HSIC table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HSIC table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SIC_SEQUENCE provided in the H7 flat file, the HSIC row will be updated.

- If a HSIC table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SIC_SEQUENCE provided in the H7 flat file does **not** exist **and** if the translator is authorized, a new HSIC row will be created.

2.7.2.8.2   Delete

To delete a SIC Code row, the H7 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and SIC_SEQUENCE are provided. A matching row owned by the translator must exist in the HSIC table.

2.7.2.9   H8 (HOTHER_PERMIT) Flat File

Environmental Permit data is fixed IOR within the context of a Region/State co-implementer pair; either the State or Region owns all of the data in this table for a given state or territory. Each Handler ID may be associated with multiple Other Permit Numbers for a given ACTIVITY_LOCATION. The translator must be IOR for HOTHER_PERMIT data to effect any changes to the table.

2.7.2.9.1   Add/Update

To add or update a Handler Other Environmental Permits table row, TRANS_CODE must = **C**.

- To update a HOTHER_PERMIT table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HOTHER_PERMIT table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** OTHER_PERMIT_NUMBER provided in the H8 flat file, the HOTHER_PERMIT row will be updated.

- If a HOTHER_PERMIT table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** OTHER_PERMIT_NUMBER provided in the H8 flat file does **not** exist **and** if the translator is authorized, a new HOTHER_PERMIT row will be created.

2.7.2.9.2 Delete

To delete an Other Environmental Permits table row, the H8 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and OTHER_PERMIT_NUMBER are provided. A matching row owned by the translator must exist in the database.

2.7.2.10 HA (HSOURCE) Flat File

HSOURCE data is variable IOR; the decision field is SOURCE_TYPE.   Each HHANDLER row for an ACTIVITY_LOCATION may have one HSOURCE row for each SOURCE_TYPE. The translator must be IOR for the specified source type for the given ACTIVITY_LOCATION in order to create or update handler information source data.

2.7.2.10.1 Add/Update

To add or update a Handler Information Source table row, TRANS_CODE must = **C**.

- To update a HSOURCE table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HSOURCE table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE provided in the HA flat file, the HSOURCE row will be updated.

- If a HSOURCE table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE provided in the HA flat file does **not** exist **and** the translator is authorized, a new HSOURCE row will be created.

2.7.2.10.2 Delete

To delete a Handler Information Source table row, the HA flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, ACTIVITY_LOCATION, and SOURCE_TYPE are provided. A matching row owned by the translator must exist in the HSOURCE table. Rows in other related flat files representing the delete Handler Information Source transaction must **NOT** be present. Deleting a Source row for a Handler ID will automatically *cascade delete* all associated rows in the Activity, and Waste Stream tables.   Related delete transactions in the HB - HE flat files will generate error messages because the data will have been previously deleted.

2.7.2.11 HB (HACTIVITY) Flat File

Handler activity data is variable IOR; the decision field is SOURCE_TYPE in the HSOURCE table. Each HSOURCE row may have one or more HACTIVITY rows associated with it. The translator must be IOR for the specified SOURCE_TYPE for the given ACTIVITY_LOCATION in order to update HACTIVITY data.

2.7.2.11.1   Add/Update

- To update a HACTIVITY table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. If a HACTIVITY table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** ACTIVITY_SEQ provided in the HB flat file, the HACTIVITY row will be updated.

- If a HACTIVITY table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** ACTIVITY_SEQ provided in the HB flat file does **not** exist **and** if the translator is authorized, a new HACTIVITY row will be created.

2.7.2.11.2   Delete

To delete a Handler Activity row, the HC flat file must contain a record where TRANS_CODE = **D,** and the key fields HANDLER_ID, ACTIVITY_LOCATION, SOURCE_TYPE, and ACTIVITY_SEQ must exist in the HACTIVITY table.

2.7.2.12   HC (HHAZARDOUS_WASTE_STREAM) Flat File

Hazardous Waste Stream data is variable IOR; the decision field is SOURCE_TYPE in the HSOURCE table. Each HSOURCE row may have multiple HHAZARDOUS_WASTE_STREAM rows associated with it. The translator must be IOR for the specified SOURCE_TYPE for the given ACTIVITY_LOCATION in order to update HHAZARDOUS_WASTE_STREAM data.

2.7.2.12.1   Add/Update

- To update a HHAZARDOUS_WASTE_STREAM table row, a replacement for the entire row must be provided.   Column-specific updates will not be supported.   If a HHAZARDOUS_WASTE_STREAM table row owned by the translator exists which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** WASTE_STREAM_SEQUENCE provided in the HC flat file, the HHAZARDOUS_WASTE_STREAM row will be updated.

2.7.2.12.1   Add/Update (continued)

- If a HHAZARDOUS_WASTE_STREAM table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** WASTE_STREAM_SEQUENCE provided in the HC flat file does **not** exist **and** the translator is authorized, a new HHAZARDOUS_WASTE_STREAM row will be created.

    Note: A HHAZARDOUS_WASTE_STREAM row should be linked to one or more HHAZARDOUS_WASTE_CODE table rows. When adding a new hazardous waste stream to the database, translators should also include corresponding waste codes in the HD (HHAZARDOUS_WASTE_CODE) flat file.

2.7.2.12.2   Delete

To delete a handler waste stream table row, the HC flat file must contain a record where TRANS_CODE = **D**, and the key fields HANDLER_ID, ACTIVITY_LOCATION, SOURCE_TYPE, and WASTE_STREAM_SEQUENCE are provided. A matching row owned by the translator must exist in

the HHAZARDOUS_WASTE_STREAM table. Rows in other related flat files representing the Delete Waste Stream transaction must **NOT** be included. Deleting a Waste Stream row for a Handler ID and Information Source will automatically *cascade delete* all associated rows in the Hazardous Waste Code and Hazardous Waste Process tables. Related delete transactions in the HD and HE flat files will generate error messages because the data will have been previously deleted.

2.7.2.13 HD (HHAZARDOUS_WASTE_CODE) Flat File

Hazardous Waste Code data is variable IOR; the decision field is SOURCE_TYPE in the HSOURCE table. Each HSOURCE row may have multiple HHAZARDOUS_WASTE_STREAM rows associated with it. Each HHAZARDOUS_WASTE_STREAM row should have one or more HHAZARDOUS_WASTE_CODE rows associated with it. The translator must be IOR for the specified SOURCE_TYPE for the given HHANDLER row ACTIVITY_LOCATION in order to update HHAZARDOUS_WASTE_CODE data.

2.7.2.13.1 Add

To add a hazardous waste code table row, TRANS_CODE must = **C**.

- If a HHAZARDOUS_WASTE_CODE table row owned by the translator which matches the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** WASTE_STREAM_SEQUENCE **plus** OWNER (of Waste Code) **plus** WASTE_CODE provided in the HD flat file does **not** exist **and** the translator is authorized, a new HHAZARDOUS_WASTE_CODE row will be created.

    > **Note: A Waste Code (WASTE_CODE) associated with a Waste Stream cannot be updated via a change transaction because it is a key column. To update an erroneous waste code, the row must be deleted and recreated with the revised key.**

2.7.2.13.2 Delete

- To delete a hazardous waste code table row, the HD flat file must contain a record where TRANS_CODE = **D**, and the key fields HANDLER_ID, ACTIVITY_LOCATION, SOURCE_TYPE, WASTE_STREAM_SEQUENCE, OWNER (of Waste Code), and WASTE_CODE are provided. A matching row owned by the translator must exist in the HHAZARDOUS_WASTE_CODE table.

2.7.2.14 HE (HHAZARDOUS_WASTE_PROCESS) Flat File

Hazardous Waste Associated Process data is variable IOR; the decision field is SOURCE_TYPE in the HSOURCE table. Each HSOURCE row may have multiple HHAZARDOUS_WASTE_STREAM rows associated with it. Each HHAZARDOUS_WASTE_STREAM row may have one or more HHAZARDOUS_WASTE_PROCESS rows associated with it. The translator must be IOR for the specified SOURCE_TYPE for a given ACTIVITY_LOCATION in order to update HHAZARDOUS_WASTE_PROCESS data.

2.7.2.14.1   Add

To add a hazardous waste associated process code table row, TRANS_CODE must = **C**.

● If a HHAZARDOUS_WASTE_CODE table row owned by the translator which matches the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** SOURCE_TYPE **plus** WASTE_STREAM_SEQUENCE **plus** OWNER (of Process Code) **plus** PROCESS_CODE provided in the HE flat file does **not** exist **and** the translator is authorized, a new HHAZARDOUS_WASTE_PROCESS row will be created.

> **Note: A Waste Associated Process (HHAZARDOUS_WASTE_PROCESS) table row for a particular Waste Stream cannot be updated via a change transaction because it is a key column. To update an erroneous process code, the row must be deleted and recreated with the revised key.**

2.7.2.14.2   Delete

To delete a hazardous waste code table row, the HE flat file must contain a record where TRANS_CODE = **D**, and the key fields HANDLER_ID, ACTIVITY_LOCATION, SOURCE_TYPE, WASTE_STREAM_SEQUENCE, OWNER (of Process Code), and PROCESS_CODE are provided. A matching row owned by the translator must exist in the HHAZARDOUS_WASTE_PROCESS table.

2.8   Permitting/Closure/Post-Closure Subject Area

A *Permitting Series* is the collection of data associated with the processing of a permit application, closure plan, permit modification, or other similar action by RCRA Permitting/Closure/Post-Closure program staff. A Treatment, Storage & Disposal Facility (TSDF) may have one or many permit applications, closure plans, etc., each of which may reference many hazardous waste process units and permitting milestone events. A Permitting Series describes a set of process unit groups at a facility, as well as the events which occur over time associated with each process unit group. A facility may request a unique application process to cover any particular unit or group of units. Under RCRA, a facility may have units which concurrently are going through closure, which have an operating permit, and which are subject to permit modification.

In RCRAInfo, each Permitting Series is assigned a *Sequence Number* and a *Series Name* which is stored in the PSERIES table. The option to supply initials for EPA and State personnel responsible for a particular Permitting Series (*EPA Person*, *State Person*, *EPA Attorney*, *State Attorney*), and *Notes* (comments) relating to the series is also available.

Each Permitting Series may be associated with many *Permitting Event* milestones. Permitting event milestone data is stored in the PEVENT table.   Because a particular event can occur more than once over time, a *Sequence Number* (which increments by *Responsible Agency*) is used as part of the key to differentiate between them. *Scheduled*, *Revised*, and *Actual Completion Dates* for the event are maintained here.   The option to provide initials for the *Responsible Person* and *Notes* about the event is available as well.

A Permitting Series may be associated with one or more *Process Unit Groups*. Process Unit Group data is stored in the PUNIT and PUNIT_DETAIL tables.   A *Sequence Number* is used as a key field to differentiate unit groups from one another. Implementers are encouraged to enter process unit group

names directly from the Part A/Part B Permit Application as supplied by the facility or make up their own to specifically describe the process unit group. A history of *Legal and Operating Statuses* for each unit group is maintained which, in conjunction with Permitting event milestone data in the PEVENT table, enables program implementers and overseers to monitor permit status. *Notes* may be optionally entered at both the unit and unit detail levels. Finally, a *linking table* which supplies the key values to link event data to unit data and conversely, unit data to event data (PLN_EVENT_UNIT) is maintained.

2.8.1 Permitting/Closure/Post-closure Subject Area Entity Relationships
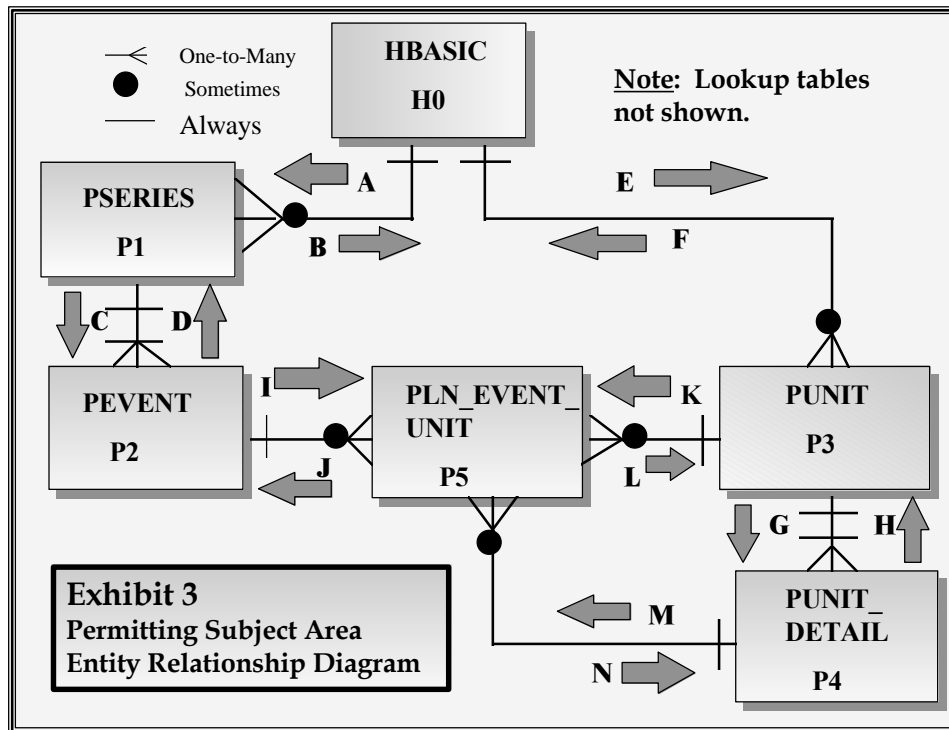
Exhibit 3 provides a graphical depiction of the entities (tables) in the Permitting subject area, and the relationships between them. Refer to the legend below for description of each relationship. Each table is directly related to a corresponding flat file. The "crows feet" indicate a one-to-many relationship between rows in two linked tables. The vertical bars indicate mandatory relationships (i.e., linkages are required). The circles indicate optional relationships (i.e., linkages are not required). Because a single Event (PEVENT) row can be associated with many Process Unit Group (PUNIT) rows, and because a single Process Unit Group (PUNIT) row can be associated with many Event (PEVENT) rows, a *Many to Many* relationship exists which must be resolved by a linking table (PLN_EVENT_UNIT).

A.   Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** Permit Series (PSERIES) rows.

B.   Each Permit Series (PSERIES) row **must always** be linked to **one** Handler Basic (HBASIC) row.

C.   Each Permit Series (PSERIES) row **must always** be linked to **one or more** Event (PEVENT) rows.

D.   Each Event (PEVENT) row **must always** be linked to **one** Permit Series (PSERIES) row.

E.   Each Handler (HBASIC) row **may or may not** be linked to **one or more** Process Unit Group (PUNIT) rows.

F.   Each Process Unit Group (PUNIT) row **must always** be linked to **one** Handler (HBASIC) row.

G.   Each Process Unit Group (PUNIT) row **must always** be linked to **one or more** Process Unit Detail (PUNIT_DETAIL) rows.

H.   Each Process Unit Detail (PUNIT_DETAIL) row **must always** be linked to **one** Process Unit (PUNIT) row.

I.   Each Event (PEVENT) row **may or may not** be linked to **one or more** Event to Unit Group Linking (PLN_EVENT_UNIT) rows.

J.   Each Event to Unit Group Linking Table (PLN_EVENT_UNIT) row **must always** be linked to **one** Event (PEVENT) row.

K.   Each Process Unit (PUNIT) row **may or may not be** be linked to **one or more** Event to Unit Group Linking (PLN_EVENT_UNIT) rows.

L.   Each Event to Unit Linking (PLN_EVENT_UNIT) row **must always** be linked to **one** Process Unit (PUNIT) row.

M.   Each Process Unit Detail (PUNIT_DETAIL) row **may or may not** be linked to **one or more** Event to Unit Group Linking   (PLN_EVENT_UNIT) rows.

N.   Each Event to Unit Group Linking (PLN_EVENT_UNIT) row **must always** be linked to **one** Process Unit Detail (PUNIT_DETAIL) row.

2.8.2   Permitting Transaction Processing

A transaction code (TRANS_CODE) must be present in each flat file for each record to indicate whether the modification is an *Add/Update* (TRANS_CODE = **C**) or a *Delete* (TRANS_CODE = **D**).

2.8.2.1   P1 (PSERIES) Flat File

PSERIES data is fixed IOR; either the State or Region owns all of the data in this table for a given state or territory (ACTIVITY_LOCATION).   PSERIES data must be owned by the translator to effect any changes to the table.

2.8.2.1.1   Add/Update

To add or update a Permit Series table row, TRANS_CODE must be = **C**.

●   To update a Permit series table row, a replacement for the entire row must be provided. Column-specific updates will not be supported.   A row matching the key columns HANDLER_ID **plus** SERIES_SEQ owned by the translator must exist in the PSERIES table.

- To create a new PSERIES row, a row matching the key columns HANDLER_ID **plus** SERIES_SEQ owned by the translator must **not** currently exist in the PSERIES table.

  Note: A PSERIES table row must be linked to one or more PEVENT table rows. When adding a new Permit Series, translators must include the event data in the P2 (PEVENT) flat file. If this data is not provided, referential integrity violations will occur.

  An Event <u>should</u> be linked to one or more PUNIT table rows. When adding a new Permit Series, translators should include the event to unit group linkage data in the P5 (PLN_EVENT_UNIT) flat file. Although RCRAInfo does not currently utilize database constraints to require mandatory linkages, the RCRAInfo data entry software will require that all Events be linked to one or more Process Units. This is to allow implementers time to clean up their data, but enforce data quality for data that is created or modified. However, database constraints may be implemented to mandate the linkages in the future.

2.8.2.1.2 Delete

In order to delete a Permit Series, the P1 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID and SERIES_SEQ are provided. Rows in other related flat files representing the delete Permit Series transaction must **NOT** be present. Deleting a Permit Series for a Handler ID will automatically *cascade delete* all associated rows in the PEVENT and PLN_EVENT_UNIT tables. Related delete transactions in the PEVENT and PLN_EVENT_UNIT flat files will cause errors because the data will have been previously deleted via the P1 flat file "Delete Permit Series" transaction.

  Note: If any entries exist in the PEVENT table for a Handler ID which have an owner other than the owner specified in the translator control file, the transaction will not be processed, and an error will result. Before deletion of the permit series can proceed, the co-implementer must separately delete all related data (i.e., any events that are associated with the permit series and related unit groups).

2.8.2.2 P2 (PEVENT) Flat File

PEVENT data is variable IOR; the decision field is *Responsible Agency*. The translator must be the responsible agency owner for the specified event for a given ACTIVITY_LOCATION in order to effect any changes to the PEVENT table.

2.8.2.2.1 Add/Update

To add or update a Permit Event, TRANS_CODE must = **C.**

- To update a Permit Event, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row matching the key columns HANDLER_ID **plus** SERIES_SEQ **plus** OWNER (of permit event code) **plus** PERMIT_EVENT_CODE **plus** ACTIVITY_LOCATION **plus** RESPONSIBLE_AGENCY **plus** EVENT_SEQUENCE_NUMBER owned by the translator must exist in the PEVENT table.

- To create a new PEVENT row, a row matching the key columns concatenated together owned by the translator must not currently exist in the PEVENT table.

2.8.2.2.2   Delete

To delete a Permit Event and all associations to a Permit Series and any Unit Groups, the P2 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, SERIES_SEQ, OWNER (of permit event code), PERMIT_EVENT_CODE, ACTIVITY_LOCATION, RESPONSIBILE_AGENCY, and EVENT_SEQUENCE_NUMBER are provided.

Rows in the P5 (PLN_EVENT_UNIT) flat file representing the Delete Permit Event transaction must NOT be present. Deleting a Permit Event row from the PEVENT table will automatically *cascade delete* any associated rows from the PLN_EVENT_UNIT table.   Related delete transactions in the PLN_EVENT_UNIT flat file will cause errors because the data will have been previously deleted.

> **Note: Deleting an event row from the PEVENT table is <u>NOT</u> the same as disassociating an event from a unit group. "Unlinking" transactions are performed via the linking P5 flat file (PLN_EVENT_UNIT).**

2.8.2.3   P3 (PUNIT) Flat File

PUNIT data is fixed IOR and should have same owner as for PSERIES data. The translator must be owner of PUNIT data for a given state or territory to effect any changes to the PUNIT table.

2.8.2.3.1   Add/Update

To add or update data for a process unit group, TRANS_CODE must = **C.**

- To update a process unit group, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row matching the key columns HANDLER_ID **plus** UNIT_SEQ owned by the translator must exist in the PUNIT table.

- To create a new PUNIT row, the key columns concatenated together must not currently exist in the PUNIT table.

> **Note: A PUNIT table row must be linked to one or more PUNIT_DETAIL table rows. When adding a new Process Unit Group, translators must include the Process Unit Detail data in the P4 (PUNIT_DETAIL) flat file.   If this data is not provided, referential integrity violations will occur.**

2.8.2.3.2   Delete

To delete a Process Unit Group and all associations to Unit Group Detail rows and Permitting Event row(s), the P3 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID and UNIT_SEQ are provided.

Rows in the P5 (PLN_EVENT_UNIT) flat file representing the Delete Process Unit Group transaction and/or P4 (PUNIT_DETAIL) flat file must **NOT** be included.   Deleting a Process Unit Group row from the PUNIT table will automatically *cascade delete* all associated rows from the PUNIT_DETAIL and PLN_EVENT_UNIT tables. Delete transactions in related flat files will cause errors because the data will have been previously deleted.

> **Note: If any entries exist in the PLN_EVENT_UNIT table for a Handler ID which have an owner other than the owner specified in the translator control file, the transaction will not be processed, and an error will result.   Before deletion of the process unit group can proceed, the co-implementer must separately delete all event to unit group linkages it owns (i.e., any events that are associated with the unit group being deleted).**

2.8.2.4   P4 (PUNIT_DETAIL) Flat File

PUNIT_DETAIL data is fixed IOR and should have same owner as for PUNIT data (and PSERIES as well). The translator must be owner of the PSERIES/PUNIT/PUNIT_DETAIL data for a given state or territory to effect any changes to the PUNIT_DETAIL table.

2.8.2.4.1   Add/Update

To add or update a process unit group detail row, TRANS_CODE must = **C.**

- To update a process unit group detail row, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row matching the key columns HANDLER_ID **plus** UNIT_SEQ **plus** UNIT_DETAIL_SEQ owned by the translator   must exist in the PUNIT_DETAIL table.

- To create a new PUNIT_DETAIL row,   the key columns concatenated together must not currently exist in the PUNIT_DETAIL table.

2.8.2.4.2   Delete

To delete a Process Unit Group Detail row and its association with the PUNIT parent row and Event row(s), the P4 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** UNIT_SEQ **plus** UNIT_DETAIL_SEQ are provided.

2.8.2.5   P5 (PLN_EVENT_UNIT) Flat File

PLN_EVENT_UNIT data is variable IOR; the decision field is *Responsible Agency* in the *PEVENT* table. The translator must be the owner of the associated PEVENT row. (The translator does not have to own the Permit Series or Process Unit Group data in order to add and maintain an event.)

> **Note: All fields in the P5 Flat File are required, and must be provided to effect a transaction.**

2.8.2.5.1   Add/Update

To add an Event to Process Unit Group link, TRANS_CODE must = **C** and key columns, which when concatenated together are unique and do not exist in the database, must be present.

> **Note: Update transactions are inappropriate for linkage tables because all columns in the table are keys which cannot be updated directly. To change a key, the row must be deleted and then recreated using the revised key.**

2.8.2.5.2   Delete

Deleting a row from the PLN_EVENT_UNIT table removes the association between an event and a process unit group. It does **not** delete the event or unit group. TRANS_CODE must = **D**, and key fields HANDLER_ID, SERIES_SEQ, OWNER (of permit event code), PERMIT_EVENT_CODE, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, EVENT_SEQ, and UNIT_SEQ must be supplied. If a match is found in the PLN_EVENT_UNIT table owned by the translator, the row will be deleted.

2.9   Corrective Action Subject Area

The Corrective Action program addresses the cleanup of hazardous waste releases at RCRA-regulated sites. States and EPA Regions provide data on the requirements imposed to address releases of hazardous waste. Sites provide States and EPA with information concerning corrective actions undertaken. The Corrective Action Subject Area of RCRAInfo is divided into three primary components: *Corrective Action Areas, Corrective Action Authorities,* and *Corrective Action Events.*

A *Corrective Action Area* refers to the area of concern for release cleanup. In RCRAInfo, a Corrective Action Area is assigned a *Sequence Number* and an *Area Name* which is stored in the AAREA table. An *Area Facility-Wide Flag* has been established to indicate that the area refers to the entire facility. A *Notes* column is used to enter a longer description of the area. The option to track the *EPA* or *State Responsible Person* is available. Columns to indicate *releases* to *groundwater*, *surface water*, *soil*, and *air* are contained in this table to be supplied as appropriate upon issuance of a Corrective Action authority.

A *Corrective Action Authority* (formerly called Corrective Action Instrument) provides the legal instrument or the programmatic basis for undertaking the corrective action process. Examples are Consent Authority, Consent Decree, Voluntary CA, Interim Status, and Policy Requirement (NCAPS ranking, stabilization evaluation). In RCRAInfo, a Corrective Action Authority is defined via its *Effective Date*, *Authority Type*, and *Responsible Agency*. Other data stored in the ACA_AUTHORITY table are *Issuance* and *Revocation Dates*, whether an *Information Repository* has been established, and entities responsible for the Authority (*Responsible Person*, *Responsible Attorney*, *Lead Program*, *Lead Suborganization*). Each Corrective Action Authority may be associated with one or more *Statutes or Citations* (formerly Legal Authority) contained in the ASTATUTE_CITATION table. Notes may be supplied for rows in both the Authority and Statute/Citation tables. Optional linkages may be created between Handler Basic (HBASIC) and CA Authority to allow an Authority row to be created before it is linked to an Event row.

Each Corrective Action Area and each Corrective Action Authority may be associated with many *Corrective Action Event* milestones. Corrective Action Event milestone data is stored in the AEVENT table. Because a particular event can occur more than once over time, a *Sequence Number* is used as part of the key to differentiate between them. Because more than one organization can establish

events, *Event Responsible Agency* is used to ensure uniqueness. *Scheduled*, *Revised*, and *Actual Completion Dates* for the event are maintained here. *Event Responsible Person* and *Suborganization*, and *Notes* may be optionally supplied. Because each event may be associated with many areas and each area may be associated with many events, a *linking table* which supplies the key values to link event data to area data and conversely, area data to event data (ALN_AREA_EVENT) is maintained.

Because each event may be associated with many authorities and each authority may be linked to many events, another linking table which supplies the key values to link event data to authority data and authority data to event data (ALN_EVENT_AUTHORITY) is required.

2.9.1   Corrective Action Subject Area Entity Relationships

Exhibit 4
Corrective Action Subject Area Entity Relationship Diagram
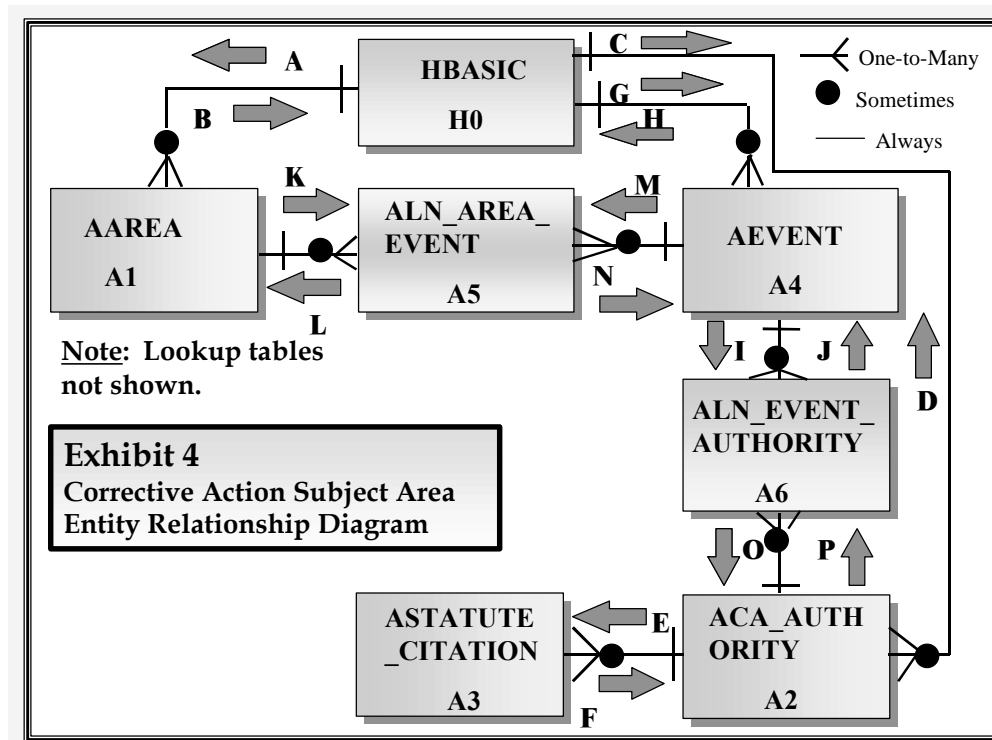
Exhibit 4 provides a graphical depiction of the entities (tables) in the Corrective Action subject area, and the relationships between them. Refer to the legend below for description of each relationship. Each table is directly related to a corresponding flat file. The "crows feet" indicate a one-to-many relationship between rows in two linked tables. The vertical bars indicate mandatory relationships (i.e., linkages are required). The circles indicate optional relationships (i.e., linkages are not required). Because a single Area (AAREA) row can be associated with many Event (AEVENT) rows, and because a single Event (AEVENT) row can be associated with many Area (AAREA) rows, a *Many to Many* relationship exists which must be resolved by a linking table (ALN_AREA_EVENT).

A.   Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CA Area (AAREA) rows.

B.   Each CA Area (AAREA) row **must always** be linked to **one** Handler Basic (HBASIC) row.

C.   Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CA Authority (ACA_AUTHORITY) rows.

D.   Each CA Authority (ACA_AUTHORITY) row **must always** be linked to **one** Handler Basic (HBASIC) row.

E.   Each CA Authority (ACA_AUTHORITY) row **may or may not** be linked to **one or more** CA Statute/Citation (ASTATUTE_CITATION) rows.

F.   Each CA Statute/Citation (ASTATUTE_CITATION) row **must always** be linked to **one** CA Authority (ACA_AUTHORITY) row.

G.   Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CA Event (AEVENT) rows.

H.   Each CA Event (AEVENT) row **must always** be linked to **one** Handler Basic (HBASIC) row.

I.   Each CA Event (AEVENT) row **may or may not** be linked to **one or more** CA Event to Authority Linking (ALN_EVENT_AUTHORITY) rows.

J.   Each CA Event to Authority Linking (ALN_EVENT_AUTHORITY) row **must always** be linked to **one** CA Event (AEVENT) row.

K.   Each CA Area (ARREA) row **may or may not** be linked to **one or more** Area to Event Linking (ALN_AREA_EVENT) rows.

L.   Each Area to Event Linking Table (ALN_AREA_EVENT) row **must always** be linked to **one** CA Event (AEVENT) row.

M.   Each CA Event (AEVENT) row **may or may not** be linked to **one or more** Area to Event Linking (ALN_AREA_EVENT) rows.

N.   Each Area to Event Linking Table (ALN_AREA_EVENT) row **must always** be linked to **one** CA Event (AEVENT) row.

O.   Each Event to Authority Linking Table (ALN_EVENT_AUTHORITY) row must always be linked to one ACA_AUTHORITY row.

P.   Each Authority (ACA_AUTHORITY) row **may or may not** be linked to **one or more** Event to Authority Linking Table (ALN_EVENT_AUTHORITY) rows.

2.9.2   Corrective Action Transaction Processing

A transaction code (TRANS_CODE) must be present in each flat file for each record to indicate whether the modification is an *Add/Update* (TRANS_CODE = **C**) or a *Delete* (TRANS_CODE = **D**).

2.9.2.1   A1 (AAREA) Flat File

AAREA data is fixed IOR; either the State or the Region owns all of the data for a given state or territory in this table. AAREA must be owned by the translator to effect any changes to the table.

2.9.2.1.1   Add/Update

To add or update an Area table row, TRANS_CODE must = **C**.

- To update an Area table row, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row matching the key columns HANDLER_ID **plus** AREA_SEQ owned by the translator must exist in the AAREA table.

- To create a new Area table row, the key columns HANDLER_ID **plus** AREA_SEQ must **not** currently exist in the AAREA table.

  Note: An AAREA table row <u>should</u> be linked to one or more AEVENT table rows. When adding a new Corrective Action Area, translators should include the area to event linkage data in the A5 (ALN_AREA_EVENT) flat file, the event data in the A4 (AEVENT) flat file, authority data in the A2 (ACA_AUTHORITY) flat file, and statute/citation data in the A3 (ASTATUTE_CITATION) flat file, and the event to authority linkage data in the A6 (ALN_EVENT_AUTHORITY) flat file.

  Although RCRAInfo does not currently utilize database constraints to require mandatory linkages, the RCRAInfo data entry software will require that all Events be linked to an Area. This is to allow implementers time to clean up their data, but enforce data quality for data that is created or modified.   However, in the future, database structure constraints will most likely be implemented to mandate these linkages. Also, be aware that the EPA Headquarters Program Accomplishments Report (PARS) select logic will ignore corrective action event data that is not linked.

2.9.2.1.2   Delete

In order to delete a Corrective Action Area, the A1 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID and AREA_SEQ are provided. Rows in other related flat files representing the Delete Area transaction must **<u>NOT</u>** be present. Deleting a Corrective Action Area for a Handler ID will automatically *cascade delete* all associated rows in the ALN_AREA_EVENT table. Related delete transactions in the ALN_AREA_EVENT table would cause the translator load process to terminate unsuccessfully because the data will have been previously deleted via the A1 flat file "Delete Corrective Action Area" transaction.

2.9.2.2   A2 (ACA_AUTHORITY) Flat File

ACA_AUTHORITY data is variable IOR; the decision field is *Authority Responsible Agency*. The translator must be the responsible agency owner for the given ACTIVITY_LOCATION for the specified authority row in order to effect any changes to the ACA_AUTHORITY table.

2.9.2.2.1   Add/Update

To add or update a Corrective Action Authority, TRANS_CODE must = **C**.

- To update a Corrective Action Authority, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** OWNER (of authority type code) **plus** AUTHORITY_TYPE **plus** ACTIVITY_LOCATION **plus** RESPONSIBLE_AGENCY (for authority) **plus** EFFECTIVE_DATE must exist in the ACA_AUTHORITY table.

- To create a new ACA_AUTHORITY row, the key columns concatenated together must **not** currently exist in the ACA_AUTHORITY table.

  Note: An ACA_AUTHORITY table row <u>should</u> be associated with one or more ASTATUTE_CITATION table rows. When adding a new ACA_AUTHORITY row, translators should include the statute/citation data in the A3 (ASTATUTE_CITATION) flat file.

2.9.2.2.2   Delete

To delete a Corrective Action Authority and associations to a statute/citation(s), the A2 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, OWNER (of authority type code), AUTHORITY_TYPE, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY (for authority), and EFFECTIVE_DATE are provided.

Rows in the A3 (ASTATUTE_CITATION) flat file representing the Delete Corrective Action Authority transaction must <u>NOT</u> be included.   Deleting a Corrective Action Authority row from the ACA_AUTHORITY table will automatically *cascade delete* any associated row(s) from the ASTATUTE_CITATION table. Related delete transactions in the ASTATUTE_CITATION flat file will cause the translator load process to terminate because the data will have been previously deleted.

2.9.2.3   A3 (ASTATUTE_CITATION) Flat File

ASTATUTE_CITATION data is variable IOR; the decision field is *Authority Responsible Agency*. The translator must be the responsible agency owner for the given ACTIVITY_LOCATION for the specified authority row in the ACA_AUTHORITY table in order to effect any changes to the ASTATUTE_CITATION table.

2.9.2.3.1   Add/Update

To add or update a Corrective Action Statute/Citation, TRANS_CODE must = **C**.

- To update a Corrective Action Authority, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** OWNER (of authority type code) **plus** AUTHORITY_TYPE **plus** ACTIVITY_LOCATION **plus** RESPONSIBLE_AGENCY (for authority) **plus** EFFECTIVE_DATE **plus** OWNER (of statutory citation code) **plus** STATUTORY_CITATION must exist in the ASTATUTE_CITATION table.

- To create a new ASTATUTE_CITATION row, the key columns concatenated together must **not** currently exist in the ASTATUTE_CITATION table.

2.9.2.4   A4 (AEVENT) Flat File

AEVENT data is variable IOR; the decision field is *Event Responsible Agency*. The translator must be the responsible agency for the specified event for the given ACTIVITY_LOCATION in order to effect any changes to the AEVENT table.

2.9.2.4.1   Add/Update

To add or update a Corrective Action Event, TRANS_CODE must = **C**.

●   To update a Corrective Action Event, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key fields HANDLER_ID **plus** OWNER (of corrective action event code) **plus** CA_EVENT_CODE **plus** ACTIVITY_LOCATION **plus** RESPONSIBLE_AGENCY **plus** EVENT_SEQ must exist in the AEVENT table.

●   To create a new AEVENT row, the key columns concatenated together must **not** currently exist in the AEVENT table.

        Note: **An AEVENT table row should be linked to a ACA_AUTHORITY table row through the ALN_EVENT_AUTHORITY linking table. An ACA_AUTHORITY row, should in turn, be linked to one or more ASTATUTE_CITATION rows. When adding a new Corrective Action Event, translators should include the authority data in the A2 (ACA_AUTHORITY) flat file, the statute/citation data in the A3 (ASTATUTE_CITATION) flat file, and link the event to an authority in the A6 (ALN_EVENT_AUTHORITY) flat file. A linkage record to an Area should also be included in the A5 (ALN_AREA_EVENT) flat file.**

        Although RCRAInfo does not currently utilize database constraints to require mandatory linkages, the RCRAInfo data entry software will require that all Events be linked to an Area. This is to allow implementers time to clean up their data, but enforce data quality for data that is created or modified. However, in the future, database structure constraints will most likely be implemented to mandate these linkages. Also, be aware that the EPA Headquarters Program Accomplishments Reports (PARS) will ignore corrective action event data that is not linked.

2.9.2.4.2   Delete

To delete a Corrective Action Event and all associations to an Area, Authorities, and Statute/Citations, the A4 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID, OWNER (of corrective action event code), CA_EVENT_CODE, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, and EVENT_SEQ are provided.

2.9.2.4.2   Delete (continued)

Rows in the A5 (ALN_AREA_EVENT) flat file representing the Delete Corrective Action Event transaction must <u>NOT</u> be present. Deleting a Corrective Action Event row from the AEVENT table will automatically *cascade delete* any associated rows from the ALN_AREA_EVENT, ACA_AUTHORITY, and ASTATUTE_CITATION tables. Related delete transactions in the associated flat files will cause the translator load process to terminate because the data will have been previously deleted.

Note: Deleting an event row from the AEVENT table is <u>NOT</u> the same as disassociating an event from an Area. "Unlinking" transactions are performed via the linking A5 flat file (ALN_AREA_EVENT).

Note: If any entries exist in the ACA_AUTHORITY table for an Event for a particular Handler ID which have an authority responsible agency owner other than the owner specified in the translator control file, the transaction will not be processed, and the translator load will not complete.   Before deletion of an Event can proceed, the co-implementer must delete all related data (i.e., any authorities that are associated with the event and related statute/citations).

2.9.2.5   A5 (ALN_AREA_EVENT) Flat File

ALN_AREA_EVENT data is variable IOR; the decision filed is *Event Responsible Agency* in the AEVENT table. The translator must be the owner of the associated AEVENT row for the given ACTIVITY_LOCATION. (The translator does not have to own the Corrective Action Area data in order to add and maintain an event.)

2.9.2.5.1   Add/Update

To add an Area to Event link, TRANS_CODE must = **C** and key columns, which when concatenated together are unique and do not exist in the database, must be present.

Note: Update transactions are inappropriate for linkage tables because all columns in the table are keys which cannot be updated directly. To change a key, the row must be deleted and then recreated using the revised key.

2.9.2.5.2   Delete

Deleting a row from the ALN_AREA_EVENT table removes the association between an area and an event. It does **not** delete the area or event. TRANS_CODE must be = **D**, and key fields HANDLER_ID, AREA_SEQ, OWNER (of corrective action event code), CA_EVENT_CODE, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, and EVENT_SEQ must be supplied.   If a match is found in the ALN_AREA_EVENT table owned by the translator, the row will be deleted.

2.9.2.6   A6 (ALN_EVENT_AUTHORITY) Flat File

ALN_EVENT_AUTHORITY data is variable IOR; the decision filed is *Event Responsible Agency* in the AEVENT table. The translator must be the owner of the associated AEVENT row for the given ACTIVITY_LOCATION. (The translator does not have to own the associated Corrective Action Authority data in order to add and maintain a linkage from an event to an authority.)

2.9.2.6.1   Add/Update

To add an Event to Authority link, TRANS_CODE must = **C** and key columns, which when concatenated together are unique and do not exist in the database, must be present.

Note: Update transactions are inappropriate for linkage tables because all columns in the table are keys which cannot be updated directly. To change a key, the row must be deleted and then recreated using the revised key.

2.9.2.6.2   Delete

Deleting a row from the ALN_EVENT_AUTHORITY table removes the association between an event and an authority. It does **not** delete the event or authority. TRANS_CODE must be = **D**, and key fields HANDLER_ID, OWNER (of corrective action event code), CA_EVENT_CODE, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY (for event), EVENT_SEQ, OWNER (of authority code), AUTHORITY_TYPE, RESPONSIBLE_AGENCY (for authority), and EFFECTIVE DATE must be supplied. If a match is found in the ALN_EVENT_AUTHORITY table owned by the translator, the row will be deleted.

2.10   Compliance Monitoring & Enforcement Subject Area

The Compliance Monitoring & Enforcement program verifies the compliance with hazardous waste both Federal and State regulations and statutes at RCRA-regulated sites. State and Regional compliance staff provide information about the completion of evaluations/inspections, the existence of violations, and enforcement activities that are being taken against hazardous waste handlers to ensure compliance with the regulations. The Compliance Monitoring & Enforcement Subject Area of RCRAInfo is divided into three primary components: *Evaluations, Violations,* and *Enforcement Actions.*

An *Evaluation* captures information about inspections and evaluations of facilities' compliance status. In RCRAInfo, a CM&E Evaluation is defined by the *Handler ID, Activity Location, Sequence Number, Evaluation Date and Agency* in the CEVALUATION table. A *Found Violation* field has been established to indicate whether or not the evaluation found any violations, or whether the agency is still determining violations. *Evaluation Type* further describes the evaluation. A *Notes* column is used to enter comments related to the evaluation. *Coverage Area* is provided to optionally specify the areas covered by the evaluation and is stored in the CEVALUATION_AREA table. The option to indicate the *Reason* for the evaluation, the *Staff* who performed the evaluation and the *Suborganization*, formally known as Branch are available.

A *Violation* indicates that a problem was discovered by an evaluation of the facility. In RCRAInfo, a Violation is defined via the *Handler ID, Activity Location*, *Sequence Number* and the *Determined By Agency*. Other data stored in the CVIOLATION table are *Determined Date*, *Violation Type*, *Citation Type*, a text field for listing the actual *Citation*, the *Actual RTC Date* (Returned to Compliance date), the *Responsible Agency*, the *Class*, and *Priority.* The option to indicate the *Staff* assigned to the violation and the *Suborganization*, formally known as Branch are available. The new field *RTC Qualifier Type* has been added to allow for the closing (returning to full physical compliance) of violations for reasons other then observed. Each Violation must be associated with one or more Evaluations via the CLN_EVALUATION_VIOLATION linking table. A *Notes* column is used to enter a comments related to the violation. Violations many have one or more *Schedule Compliance Date* ordered by one or more enforcement action, this date and link is stored in the CLN_COMPLIANCE_SCHEDULE linking table.

An *Enforcement* addresses violations which were discovered at the facility.   In RCRAInfo, an Enforcement is defined by: *Handler ID, Activity Location*, *Sequence Number, Enforcement Date* and *Agency*. Other data stored in the CENFORCEMENT table are *Docket* number, *Enforcement Type* and *Attorney*. The option to indicate the *Staff* assigned to the violation and the *Suborganization*, formally known as Branch are available. Enforcement actions may a *Schedule Compliance* date which is stored in the CLN_COMPLIANCE_SCHEDULE linking table.   Multiple penalties can be assessed by an enforcement action and are stored in the CPENALTY table. The CPENALTY table is defined *Penalty Type* and stores *Amount* and *Notes* related to the penalty. Payments are paid against penalties, therefore CPAYMENT is always defined by one or more CPENALTY. The CPAYMENT table is defined

by *Sequence Number* and stores the *Scheduled Date* and *Scheduled Amount* of the payment, and the *Paid Date* and *Paid Amount* of the payment. It also provided a *Notes* field to enter comments related to the payment. Since an enforcement action may involve other media a CMEDIA table provided the opportunity to record *Media Type* and *Notes*. Supplemental Environmental Projects (SEPs) may be ordered by the enforcement action, the CSEP table stores this information. SEPs are defined by Sequence Number. Other information stored related to SEPs is *SEP Type*, *Expenditure Amount*, *Scheduled Date*, *Actual Date*, *Staff*, and Notes relating to the SEP.

State and Regional compliance staff provide information about the completion of inspections, the existence of violations of RCRA or State regulations, and enforcement activities that are being taken against hazardous waste handlers to ensure compliance with the regulations.

2.10.1 Compliance Monitoring & Enforcement Subject Area Entity Relationships

Exhibit 5
Compliance Monitoring& Enforcement
Entity Relationship Diagram
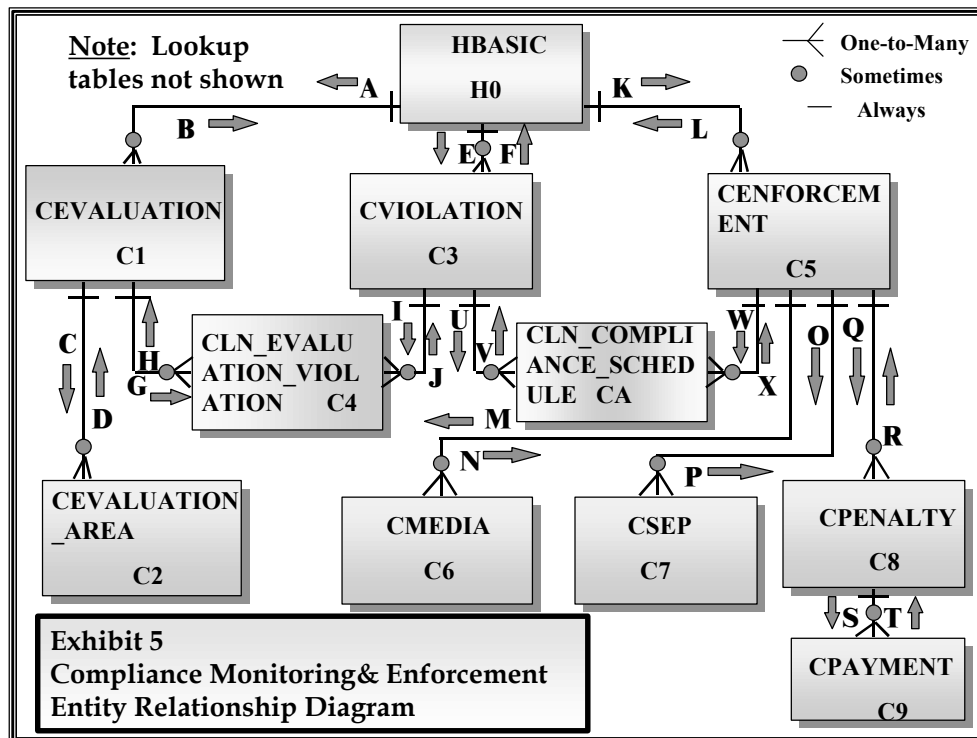
Exhibit 5 provides a graphical depiction of the entities (tables) in the Compliance Monitoring & Enforcement subject area, and the relationships between them. Refer to the legend below for a description of each relationship. Each table is directly related to a corresponding flat file. The "crows feet" indicate a one-to-many relationship between rows in two linked tables. The vertical bars indicate mandatory relationships (i.e., linkages are required). The circles indicate optional relationships (i.e., linkages are not required).

A.   Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CM&E Evaluation (CEVALUATION) rows.

B.   Each CM&E Evaluation (CEVALUATION) row **must always** be linked to **one** Handler Basic (HBASIC) row.

C. Each CM&E Evaluation (CEVALUATION) row **may or may not** be linked to **one or more** CM&E Evaluation Area (CEVALUATION_AREA) rows.

D. Each CM&E Evaluation Area (CEVALUATION_AREA) row **must always** be linked to **one** CM&E Evaluation (CEVALUATION).

E. Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CM&E Violation (CVIOLATION) rows.

F. Each CM&E Violation (CVIOLATION) row **must always** be linked to **one** Handler Basic (HBASIC) row.

G. Each CM&E Evaluation (CEVALUATION) row **may or may not** be linked to **one or more** CM&E Evaluation to Violation linking Table (CLN_EVALUATION_VIOLATION) rows.

H. Each CM&E Evaluation to Violation linking Table (CLN_EVALUATION_VIOLATION) row **must always** be linked to **one** CM&E Evaluation (CEVALUATION) row.

I. Each CM&E Violation (CVIOLATION) row **may or may not** be linked to **one or more** CM&E Evaluation to Violation linking Table (CLN_EVALUATION_VIOLATION) rows.

J. Each CM&E Evaluation to Violation linking Table (CLN_EVALUATION_VIOLATION) row **must always** be linked to **one** CM&E Violation (CVIOLATION) row.

K. Each Handler Basic (HBASIC) row **may or may not** be linked to **one or more** CM&E Enforcement (CENFORCEMENT) rows.

L. Each CM&E Enforcement (CENFORCEMENT) row **must always** be linked to **one** Handler Basic (HBASIC) row.

M. Each CM&E Enforcement (CENFORCEMENT) row **may or may not** be linked to **one or more** CM&E Multimedia (CMEDIA) rows.

N. Each CM&E Multimedia (CMEDIA) row **must always** be linked to **one** CM&E Enforcement (CENFORCEMENT) row.

O. CM&E Enforcement (CENFORCEMENT) row **may or may not** be linked to **one or more** CM&E SEP (CSEP) rows.

P. Each CM&E SEP (CSEP) row **must always** be linked to **one** CM&E Enforcement (CENFORCEMENT) row.

Q. CM&E Enforcement (CENFORCEMENT) row **may or may not** be linked to **one or more** CM&E Penalty (CPENALTY) rows.

R. Each CM&E Penalty (CPENALTY) row **must always** be linked to **one** CM&E Enforcement (CENFORCEMENT) row.

S.   CM&E Penalty (CPENALTY) row **may or may not** be linked to **one or more** CM&E Payment (CPAYMENT) rows.

T.   Each CM&E Payment (CPAYMENT) row **must always** be linked to **one** CM&E Penalty (CPENALTY) row.

2.10.2 Compliance Monitoring & Enforcement Transaction Processing

A transaction code (TRANS_CODE) must be present in each flat file for each record to indicate whether the modification is an *Add/Update* (TRANS_CODE = **C**) or a *Delete* (TRANS_CODE = **D**).

2.10.2.1   C1 (CEVALUATION) Flat File

CEVALUATION data is variable IOR; the decision field is *Agency*.   The translator must be the responsible agency owner for the given ACTIVITY_LOCATION for the specified evaluation row in order to effect any changes to the CEVALUATION table.

2.10.2.1.1   Add/Update

To add or update an Evaluation, TRANS_CODE must = **C**.

●   To update an Evaluation, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EVALUATION_DATE **plus** EVALUATION_SEQ **plus** AGENCY must exist in the CEVALUATION table.

●   To create a new CEVALUATION row, the key columns concatenated together must **not** currently exist in the CEVALUATION table.

2.10.2.1.2   Delete

To delete a CM&E Evaluation, the C1 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EVALUATION_DATE **plus** EVALUATION_SEQ **plus** AGENCY are provided.

Rows in the C2 (CEVALUATION_AREA) flat file, the C3 (CVIOLATION) flat file, the C4 (CLN_EVALUATION_VIOLATION) flat file, the CA (CLN_COMPLIANCE_SCHEDULE) flat file, the C5 (CENFORCEMENT) flat file, the C6 (CMEDIA) flat file, the C7 (CSEP) flat file, the C8 (CPENALTY) flat file and C9 (CPAYMENT) flat file representing the Delete CM&E Evaluation transaction must <u>NOT</u> be included. Deleting a CM&E Evaluation row from the CEVALUATION table will automatically *cascade delete* any associated row(s) from the CEVALUATION_AREA table, the CVIOLATION table, the CLN_EVALUATION_VIOLATION linking table, the CLN_COMPLIANCE_SCHEDULE linking table, the CENFORCEMENT table, the CMEDIA table, the CSEP table, the CPENALTY table and the CPAYMENT table. Related delete transactions in those flat files will cause the translator load process to terminate because the data will have been previously deleted.

2.10.2.2   C2 (CEVALUATION_AREA) Flat File

CEVALUATION_AREA data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified evaluation area row in order to effect any changes to the CEVALUATION_AREA table.

2.10.2.2.1   Add/Update

To add or update an Evaluation Area, TRANS_CODE must = **C**.

●   To update an Evaluation Area, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** OWNER (of area type code) **plus** AREA_TYPE **plus** ACTIVITY_LOCATION **plus** AGENCY **plus** EVALUATION_DATE **plus** EVALUATION_SEQ must exist in the CEVALUATION_AREA table.

●   To create a new CEVALUATION_AREA row, the key columns concatenated together must **not** currently exist in the CEVALUATION_AREA table.

2.10.2.2.2   Delete

To delete a CM&E Evaluation Area, the C2 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** OWNER (of area type code) **plus** AREA_TYPE **plus** ACTIVITY_LOCATION **plus** AGENCY **plus** EVALUATION_DATE **plus** EVALUATION_SEQ are provided.

Rows in the C3 (CVIOLATION) flat file, the C4 (CLN_EVALUATION_VIOLATION) flat file, the CA (CLN_COMPLIANCE_SCHEDULE) flat file, the C5 (CENFORCEMENT) flat file, the C6 (CMEDIA) flat file, the C7 (CSEP) flat file, the C8 (CPENALTY) flat file and C9 (CPAYMENT) flat file representing the Delete CM&E Evaluation Area transaction must <u>NOT</u> be included. Deleting a CM&E Evaluation Area row from the CEVALUATION_AREA table will automatically *cascade delete* any associated row(s) from the CVIOLATION table, the CLN_EVALUATION_VIOLATION linking table, the CLN_COMPLIANCE_SCHEDULE linking table, the CENFORCEMENT table, the CMEDIA table, the CSEP table, the CPENALTY table and the CPAYMENT table. Related delete transactions in those flat files will cause the translator load process to terminate because the data will have been previously deleted.

2.10.2.3   C3 (CVIOLATION) Flat File

CVIOLATION data is variable IOR; the decision field is *Responsible Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CVIOLATION table.

2.10.2.3.1   Add/Update

To add or update a Violation, TRANS_CODE must = **C**.

- To update a Violation, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** DETERMINED BY AGENCY **plus** VIOLATION_SEQ must exist in the CVIOLATION table.

- To create a new CVIOLATION row, the key columns concatenated together must **not** currently exist in the CVIOLATION table.

   Note: **A CVIOLATION table row <u>must</u> be linked to one or more CEVALUATION table rows. When adding a new Violation, translators must include the Evaluation data in the C1 (CEVALUATION) flat file and the linkage record to the Evaluation <u>must</u> also be included in the C4 (CLN_EVALUATION_VIOLATION) flat file.**

   RCRAInfo will utilize database constraints to require mandatory linkages.

2.10.2.3.2   Delete

To delete a CM&E Violation, the C3 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** DETERMINED BY AGENCY **plus** VIOLATION_SEQ are provided.

Rows in the C4 (CLN_EVALUATION_VIOLATION) flat file, the CA (CLN_COMPLIANCE_SCHEDULE) flat file, the C5 (CENFORCEMENT) flat file, the C6 (CMEDIA) flat file, the C7 (CSEP) flat file, the C8 (CPENALTY) flat file and C9 (CPAYMENT) flat file representing the Delete CM&E Violation transaction must <u>NOT</u> be included. Deleting a CM&E Violation row from the CVIOLATION table will automatically *cascade delete* any associated row(s) from the CLN_EVALUATION_VIOLATION linking table, the CLN_COMPLIANCE_SCHEDULE linking table, the CENFORCEMENT table, the CMEDIA table, the CSEP table, the CPENALTY table and the CPAYMENT table. Related delete transactions in those flat files will cause the translator load process to terminate because the data will have been previously deleted.

2.10.2.4   C4 (CLN_EVALUATION_VIOLATION) Flat File

CLN_EVALUATION_VIOLATION data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CLN_EVALUATION_VIOLATION table.

2.10.2.4.1   Add/Update

To add or update the Evaluation to Violation linking file, TRANS_CODE must = **C**.

- To update the Evaluation to Violation linking file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EVALUATION_DATE **plus** EVALUATION_SEQ **plus** AGENCY **plus** DETERMINED BY AGENCY **plus** VIOLATION_SEQ must exist in the CLN_EVALUATION_VIOLATION table.

- To create a new CLN_EVALUATION_VIOLATION row, the key columns concatenated together must **not** currently exist in the CLN_EVALUATION_VIOLATION table.

2.10.2.4.2   Delete

To delete a CM&E Evaluation to Violation linking file, the C4 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** EVALUATION_DATE **plus** EVALUATION_SEQ **plus** AGENCY **plus** DETERMINED BY AGENCY **plus** VIOLATION_SEQ are provided.

2.10.2.5   C5 (CENFORCEMENT) Flat File

CENFORCEMENT data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CENFORCEMENT table.

2.10.2.5.1   Add/Update

To add or update the Enforcement file, TRANS_CODE must = **C**.

- To update the Enforcement file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY must exist in the CENFORCEMENT table.

- To create a new CENFORCEMENT row, the key columns concatenated together must **not** currently exist in the CENFORCEMENT table.

2.10.2.5.2   Delete

To delete a CM&E Enforcement file, the C5 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY are provided.

Rows in the CA (CLN_COMPLIANCE_SCHEDULE) flat file, the C6 (CMEDIA) flat file, the C7 (CSEP) flat file, the C8 (CPENALTY) flat file and C9 (CPAYMENT) flat file representing the Delete CM&E Enforcement transaction must NOT be included. Deleting a CM&E Enforcement row from the CENFORCEMENT table will automatically *cascade delete* any associated row(s) from the CLN_COMPLIANCE_SCHEDULE table, the CMEDIA table, the CSEP table, the CPENALTY table and the CPAYMENT table. Related delete transactions in those flat files will cause the translator load process to terminate because the data will have been previously deleted.

2.10.2.6   C6 (CMEDIA) Flat File

CMEDIA data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CMEDIA table.

2.10.2.6.1   Add/Update

To add or update the Multimedia file, TRANS_CODE must = **C**.

● To update the Multimedia file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** MEDIA TYPE must exist in the CMEDIA table.

● To create a new CMEDIA row, the key columns concatenated together must **not** currently exist in the CMEDIA table.

2.10.2.6.2   Delete

To delete a CM&E Multimedia file, the C6 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** MEDIA TYPE are provided.

2.10.2.7   C7 (CSEP) Flat File

CSEP data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CSEP table.

2.10.2.7.1   Add/Update

To add or update the SEP file, TRANS_CODE must = **C**.

● To update the SEP file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** SEP SEQ must exist in the CSEP table.

● To create a new CSEP row, the key columns concatenated together must **not** currently exist in the CSEP table.

2.10.2.7.2   Delete

To delete a CM&E SEP file, the C7 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** SEP SEQ are provided.

2.10.2.8   C8 (CPENALTY) Flat File

CPENALTY data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CPENALTY table.

2.10.2.8.1   Add/Update

To add or update the Penalty file, TRANS_CODE must = **C**.

- To update the Penalty file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** PENALTY TYPE must exist in the CPENALTY table.

- To create a new CPENALTY row, the key columns concatenated together must **not** currently exist in the CPENALTY table.

2.10.2.8.2   Delete

To delete a CM&E Penalty file, the C8 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** PENALTY TYPE are provided.

Rows in the C9 (CPAYMENT) flat file representing the Delete CM&E Penalty transaction must <u>NOT</u> be included. Deleting a CM&E Penalty row from the CPENALTY table will automatically *cascade delete* any associated row(s) from the CPAYMENT table. Related delete transactions in that flat file will cause the translator load process to terminate because the data will have been previously deleted.

2.10.2.9   C9 (CPAYMENT) Flat File

CPAYMENT data is variable IOR; the decision field is *Agency*. The translator must be the agency owner for the given ACTIVITY_LOCATION for the specified violation row in order to effect any changes to the CPAYMENT table.

2.10.2.9.1   Add/Update

To add or update the Payment file, TRANS_CODE must = **C**.

- To update the Payment file, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** PENALTY TYPE **plus** PAYMENT SEQ must exist in the CPAYMENT table.

- To create a new CPAYMENT row, the key columns concatenated together must **not** currently exist in the CPAYMENT table.

2.10.2.9.2   Delete

To delete a CM&E Payment file, the C9 flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** PENALTY TYPE **plus** PAYMENT SEQ are provided.

2.10.2.10   CA (CLN_COMPLIANCE_SCHEDULE) Flat File

CLN_COMPLIANCE_SCHEDULE data is variable IOR; the decision field is *Agency*. The translator must be the responsible agency owner for the given ACTIVITY_LOCATION for the specified evaluation row in order to effect any changes to the CLN_COMPLIANCE_SCHEDULE table.

2.10.2.10.1   Add/Update

To add or update a Compliance Schedule link, TRANS_CODE must = **C**.

- To update a Compliance Schedule link, a replacement for the entire row must be provided. Column-specific updates will not be supported. A row owned by the translator matching the key columns HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** DETERMINED_BY_AGENCY **plus** VIOLATION_SEQ must exist in the CLN_COMPLIANCE_SCHEDULE linking table.

- To create a new CLN_COMPLIANCE_SCHEDULE row, the key columns concatenated together must **not** currently exist in the CLN_COMPLIANCE_SCHEDULE table.

2.10.2.10.2   Delete

To delete a CM&E Compliance Schedule link, the CA flat file must contain a record where TRANS_CODE = **D** and the key fields HANDLER_ID **plus** ACTIVITY_LOCATION **plus** ENFORCEMENT_DATE **plus** ENFORCEMENT_SEQ **plus** AGENCY **plus** DETERMINED_BY_AGENCY **plus** VIOLATION_SEQ are provided.

Rows in the C5 (CENFORCEMENT) flat file, the C6 (CMEDIA) flat file, the C7 (CSEP) flat file, the C8 (CPENALTY) flat file and C9 (CPAYMENT) flat file representing the Delete CM&E Compliance Schedule link transaction must <u>NOT</u> be included. Deleting a CM&E Compliance Schedule link row from the CLN_COMPLIANCE_SCHEDULE link table will automatically *cascade delete* any associated row(s) from the CENFORCEMENT table, the CMEDIA table, the CSEP table, the CPENALTY table and the CPAYMENT table. Related delete transactions in those flat files will cause the translator load process to terminate because the data will have been previously deleted.

APPENDIX A: Flat File Specifications

FLAT FILE NAMING CONVENTIONS

Flat file names are constructed in the following manner:

SSFFNNNs.FIL

Where:

SS	=	State Postal Code
FF	=	Flat File Identifier (eg., P1 or P2)
NNN	=	Julian Date when file was created.
s	=	Sequence indicator for Flat File

Note: The two-character Flat File Identifier distinguishes each flat file produced during the translation. For example, the correct name for the first P1 file, containing AL data, produced on January 4th, is ALP1004A.FIL.

FLAT FILE ID# - CL

Table Name:	All				Record Length: 28			
Description:	Control File - Name & statistical information for each flat file submitted.							

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	FLATFILE_NAME	N	Text	12		N/A	Y	CL010, CL012, CL014, CL016
2	13	SOFTWARE_VERSION	N	Text	1		N/A	Y	CL020
3	14	DATE_CREATED	N	CCYYMMDD	8		N/A	Y	CL030
4	22	RECORD_COUNT	N	Integer	6		N/A	Y	CL040
5	28	FULL_REPLACE_IND	N	Text	1		N/A	Y**	CL050, CL052

NOTES: All text data in the Control File must be in upper case.
ACTIVITY_LOCATION and identity of translator will be derived from translator load job submittal process.

FLAT FILE ID# - H0

Table Name:	HBASIC					IOR Type:	Fixed		
						Record Length:	306		
Description:	Handler Basic Identification								

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID	Y	Text	12	1	Y*	Y	H0000, H0002, H0010, H0012, H0014, H0016, H0018
3	14	HANDLER_NAME	N	Text	40		Y*	Y	H0020
4	54	EXTRACT_FLAG	N	Text	1		Y*	Y	H0030
5	55	FACILITY_IDENTIFIER	N	Text	12		Y*	N	--
6	67	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG = **'X'.**
**   Required for transaction processing.

FLAT FILE ID# - H1

Table Name:	HHANDLER				IOR Type:	Fixed		
Description:	Handler Identification Data				Record Length:	530		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	H1010, H1012, H1014, H1016, H1018
3	14	ACTIVITY_LOCATION	Y	Text	2	2	Y*	Y	H1000, H1010, H1014, H1016, H1018
4	16	OWNER (*FK_LU_COUNTYOWNER*)	N	Text	2		Y*	Y	H1020, H1021
5	18	COUNTY_CODE (*FK_LU_COUNTYCOUNTY_CODE*)	N	Text	5		Y*	Y	H1021
6	23	LOCATION_STREET_NO	N	Text	12		Y*	N	--
7	35	LOCATION_STREET1	N	Text	30		Y*	Y	H1030
8	65	LOCATION_STREET2	N	Text	30		Y*	N	--
9	95	LOCATION_CITY	N	Text	25		Y*	Y	H1032
10	120	LOCATION_STATE	N	Text	2		Y*	Y	H1034
11	122	LOCATION_ZIP	N	Text	9		Y*	Y	H0136
12	131	MAIL_STREET_NO	N	Text	12		Y*	N	--
13	143	MAIL_STREET1	N	Text	30		Y*	Y***	H1040
14	173	MAIL_STREET2	N	Text	30		Y*	N	--
15	203	MAIL_CITY	N	Text	25		Y*	Y***	H1042
16	228	MAIL_STATE	N	Text	2		Y*	Y***	H1044
17	230	MAIL_ZIP	N	Text	9		Y*	Y***	H1046
18	239	LAND_TYPE	N	Text	1		Y*	N	H1050
19	240	PREVIOUS_ID	N	Text	12		Y*	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.
***   Required if NON_NOTIFIER in HHANDLER (H1) NE **'X'** or **'E'**

| Table Name: | HHANDLER | | | IOR Type: | Fixed | | | | |
| Description: | Handler Identification Data | | | Record Length: 530 | | | | | |

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
20	252	NON_NOTIFIER	N	Text	1		Y*	N	H1060
21	253	ACCESSIBILITY	N	Text	1		Y*	N	H1070
22	254	TSD_DATE	N	CCYYMMDD	8		Y*	N	H1080
23	262	OFF_SITE_RECEIPT	N	Text	1		Y*	N	H1090
24	263	SECOND_ID	N	Text	12		N	Y**	H1100
25	275	STATE_DISTRICT	N	Text	10		N	N	--
26	285	RIVER_BASIN	N	Text	6		N	N	--
27	291	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required if HANDLER_ID does not pass algorithm check.

FLAT FILE ID# - H2

| Table Name: | HPREVIOUS_NAME | | IOR Type: | Fixed |
| Description: | Handler Previous Name Information | | Record Length: | 303 |

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	H2010, H2012, H2016
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y	H2000, H2016
4	16	RECEIVE_DATE	Y	CCYYMMDD	8D	3-D	Y*	Y	H2014, H2016, H2018
5	24	PREVIOUS_HANDLER_NAME	N	Text	40		Y*	Y	H2020
6	64	NOTES	N	Text	240		N	N	--

* Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = '**X**'.
** Required for transaction processing.

FLAT FILE ID# - H3

Table Name:	HLATITUDE_LONGITUDE				IOR Type:	Fixed		
Description:	Handler Coordinate Information				Record Length: 271			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y***	H3010, H3012, H3014, H3016
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y***	H3000, H3010, H3012, H3014, H3016
4	16	LATITUDE_LONGITUDE_SOURCE	N	Text	1		Y*	Y***	H3020
5	17	LATITUDE_MEASURE	N	Text	7		Y*	Y***	H3030
6	24	LONGITUDE_MEASURE	N	Text	8		Y*	Y***	H3040
7	32	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.
***   Data should be provided when submitted by the handler (LATITUDE_LONGITUDE_SOURCE = '**A**').

FLAT FILE ID# - H4

Table Name:	HENV_PRIORITY_RANK				IOR Type:	Fixed		
Description:	Handler Environmental Priority Ranking				Record Length:	264		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys.Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	N	Y	H4010, H4012, H4016, H4018
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	N	Y	H4000, H4010, H4012, H4016, H4018
4	16	EPR_DATE	Y	CCYYMMDD	8	3-D	N	Y	H4014, H4018
5	24	EPR_STATUS	N	Text	1		N	Y	H4020
6	25	NOTES	N	Text	240		N	N	--

*   Required for transaction processing.

FLAT FILE ID# - H5

Table Name:	HCONTACT		IOR Type:	Variable
Description:	Handler Contact Information		Record Length:	414
			Decision Field:	CONTACT_TYPE

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y***	H5020, H5022, H5024, H5026
3	14	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y***	H5000, H5020, H5022, H5024, H5026
4	16	OWNER (*FK_LU_CONTACTOWNER*)	Y	Text	2		Y*	Y***	H5010, H5011, H5020, H5024, H5026
5	18	CONTACT_TYPE (*FK_LU_CONTACTCONTACT_TYPE*)	Y	Text	1	3	Y*	Y***	H5011, H5020, H5024, H5026
6	19	CONTACT_TITLE	N	Text	15		Y*	N	--
7	34	CONTACT_LAST_NAME	N	Text	15		Y*	Y***	H5030
8	49	CONTACT_FIRST_NAME	N	Text	15		Y*	N	--
9	64	STREET1	N	Text	30		Y*	Y***	H5040
10	94	STREET2	N	Text	30		Y*	N	--
11	124	CITY	N	Text	30		Y*	Y***	H5042
12	154	STATE	N	Text	2		Y*	Y***	H5044
13	156	ZIP	N	Text	9		Y*	Y***	H5046
14	165	PHONE	N	Text	10		Y*	N	--
15	175	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X' and** CONTACT_TYPE = **'N'**, **'A'**, or **'R'.**
**   Required for transaction processing.
***   Required if NON_NOTIFIER in HHANDLER (H1) NE '**X**' or **'E' and** CONTACT_TYPE = '**N'** or '**A'**.

FLAT FILE ID# - H6

Table Name:	HOWNER_OPERATOR			IOR Type:	Fixed			
Description:	Handler Owner & Operator Information			Record Length: 417				

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	H6010, H6012, H6016, H6018
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y	H6000, H6010, H6012, H6016, H6018
4	16	OWNER_OPERATOR_SEQ	Y	Integer	4	3	Y*	Y	H6014, H6016, H6018
5	20	OWNER_OPERATOR_INDICATOR	N	Text	2		Y*	Y	H6020
6	22	OWNER_OPERATOR_TYPE	N	Text	2		Y*	Y	H6030
7	24	OWNER_OPERATOR_CHANGE_DATE	N	CCYYMMDD	8		Y*	Y***	H6040, H6042
8	32	OWNER_OPERATOR_NAME	N	Text	40		Y*	Y	H6050
9	72	PHONE	N	Text	10		Y*	Y	H6060
10	82	STREET1	N	Text	30		Y*	Y	H6070
11	112	STREET2	N	Text	30		Y*	N	--
12	142	CITY	N	Text	25		Y*	Y	H6072
13	167	STATE	N	Text	2		Y*	Y	H6074
14	169	ZIP	N	Text	9		Y*	Y	H6076
15	178	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.
***   Required if OWNER_OPERATOR_INDICATOR = '**PO'** or **'PP'**.

FLAT FILE ID# - H7

Table Name:	HSIC				IOR Type:	Fixed		
Description:	Handler SIC Code Data				Record Length: 27			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	H7010, H7012, H7016, H7018
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y	H7000, H7010, H7012, H7016,H7018
4	16	SIC_SEQUENCE	Y	Integer	4	3	Y*	Y	H7114, H7016, H7018
5	20	OWNER (*FK_LU_SICOWNER*)	N	Text	2		Y*	Y	H7020, H7021
6	22	SIC_CODE (*FK_LU_SICSIC_CODE*)	N	Text	4		Y*	Y	H7021, H7020
7	26	PRIMARY_SIC_INDICATOR	N	Text	1		Y*	Y	H7030
8	27	SIC_SOURCE	N	Text	1		Y*	Y	H7040

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.

FLAT FILE ID# - H8

Table Name:	HOTHER_PERMIT			IOR Type:	Fixed		
Description:	Handler Other Environmental Permits			Record Length: 291			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052
2	2	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	N	Y	H8010, H8012, H8016, H8018
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	N	Y	H8000, H8010, H8012, H8016, H8018
4	16	OTHER_PERMIT_NUMBER	Y	Text	13	3	N	Y	H8014, H8016, H8018
5	29	OWNER (*FK_LU_OTHER_PEROWNER*)	N	Text	2		N	Y	H8020, H8021
6	31	OTHER_PERMIT_TYPE (*FK_LU_OTHER_PEROTHER_PERMIT_TY*)	N	Text	1		N	Y	H8021, H8020
7	32	OTHER_PERMIT_DESC	N	Text	20		N	N	--
8	52	NOTES	N	Text	240		N	N	--

*   Required for transaction processing.

FLAT FILE ID# - HA

Table Name: HSOURCE Description: Source of Activity & Waste Stream Information						IOR Type: Variable Decision Field: SOURCE_TYPE Record Length: 496			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FKHHANDLERFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	HA012, HA014, HA016, HA018
3	14	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Y	Text	2	2	Y*	Y	HA000, HA012, HA014, HA016, HA018
4	16	SOURCE_TYPE	Y	Text	1	3	Y*	Y	HA010, HA016, HA018
5	17	NOTES	N	Text	480		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.

FLAT FILE ID# - HB

<table>
<tr><td colspan="5">Table Name: HACTIVITY
Description: Handler Activity Data</td><td colspan="5">IOR Type: Variable
Decision Field: SOURCE_TYPE
Record Length: 407</td></tr>
<tr><th>No.</th><th>Pos.</th><th>Field Name</th><th>Key</th><th>Type</th><th>Size</th><th>Sort</th><th>Nat. Req.</th><th>Sys. Req.</th><th>Edit #</th></tr>
<tr><td>1</td><td>1</td><td>TRANS_CODE</td><td>N</td><td>Text</td><td>1</td><td></td><td>N/A</td><td>Y***</td><td>CL050, CL052</td></tr>
<tr><td>2</td><td>2</td><td>HANDLER_ID (<i>FK_HSOURCEFK_HHANDLERFK_HBASIC</i>)</td><td>Y</td><td>Text</td><td>12</td><td>1</td><td>Y*</td><td>Y</td><td>HB012, HB014, HB018, HB020</td></tr>
<tr><td>3</td><td>14</td><td>ACTIVITY_LOCATION (<i>FK_HSOURCEFK_HHANDLERACTIVITY</i>)</td><td>Y</td><td>Text</td><td>2</td><td>2</td><td>Y*</td><td>Y</td><td>HB000, HB012, HB014, HB018, HB020</td></tr>
<tr><td>4</td><td>16</td><td>SOURCE_TYPE (<i>FK_HSOURCESOURCE_TYPE</i>)</td><td>Y</td><td>Text</td><td>1</td><td>3</td><td>Y*</td><td>Y</td><td>HB010, HB012, HB014, HB018, HB020</td></tr>
<tr><td>5</td><td>17</td><td>ACTIVITY_SEQ</td><td>Y</td><td>Number</td><td>4</td><td>4</td><td>Y*</td><td>Y</td><td>HB016, HB018, HB020</td></tr>
<tr><td>6</td><td>21</td><td>RECEIVE_DATE</td><td>N</td><td>CCYYMMDD</td><td>8</td><td></td><td>Y*</td><td>Y</td><td>HB030</td></tr>
<tr><td>7</td><td>29</td><td>OWNER (<i>FK_LU_GENERATOROWNER</i>)</td><td>N</td><td>Text</td><td>2</td><td></td><td>Y*</td><td>Y</td><td>HB040, HB041</td></tr>
<tr><td>8</td><td>31</td><td>GENERATOR_ACTIVITY (<i>FK_LU_GENERATORGENERATOR_ACTIV</i>)</td><td>N</td><td>Text</td><td>1</td><td></td><td>Y**</td><td>Y</td><td>HB041, HB190</td></tr>
<tr><td>9</td><td>32</td><td>GENERATOR_RCRA_STATUS</td><td>N</td><td>Text</td><td>1</td><td></td><td>Y**</td><td>N</td><td>HB042</td></tr>
<tr><td>10</td><td>33</td><td>GENERATOR_RCRA_DESC</td><td>N</td><td>Text</td><td>2</td><td></td><td>Y**</td><td>N</td><td>HB044</td></tr>
</table>

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X' and** GENERATOR_ACTIVITY = '**1**', '**2**' or **'N'.**
***   Required for transaction processing.

Table Name: HACTIVITY	IOR Type: Variable
Description: Handler Activity Data	Decision Field: SOURCE_TYPE
	Record Length: 407

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
11	35	GENERATOR_STATE_STATUS	N	Text	1		N	N	--
12	36	GENERATOR_STATE_DESC	N	Text	12		N	N	--
13	48	TRANSPORTER_ACTIVITY	N	Text	1		Y*	Y	HB050, HB056, HB057, HB058, HB190
14	49	AIR	N	Text	1		Y*	Y**	HB051, HB058
15	50	RAIL	N	Text	1		Y*	Y**	HB052, HB058
16	51	HIGHWAY	N	Text	1		Y*	Y**	HB053, HB058
17	52	WATER	N	Text	1		Y*	Y**	HB054, HB058
18	53	OTHER_MODE	N	Text	20		Y*	Y**	HB058
19	73	TRANSPORTER_RCRA_STATUS	N	Text	1		Y*	Y***	HB056, HB057, HB058
20	74	TRANSPORTER_RCRA_DESC	N	Text	2		Y*	N	HB059
21	76	TRANSPORTER_STATE_STATUS	N	Text	1		N	N	--
22	77	TRANSPORTER_STATE_DESC	N	Text	12		N	N	--
23	89	TSD_ACTIVITY	N	Text	1		Y*	Y	HB060, HB062, HB064

FLAT FILE ID# - HB (continued)

Table Name: HACTIVITY Description: Handler Activity Data			IOR Type: Variable Decision Field: SOURCE_TYPE Record Length: 407						
24	90	TSD_RCRA_STATUS	N	Text	1		Y*	Y****	HB062, HB064, HB190
25	91	TSD_RCRA_DESC	N	Text	2		Y*	N	HB066
26	93	TSD_STATE_STATUS	N	Text	1		N	N	--
27	94	TSD_STATE_DESC	N	Text	12		N	N	--
28	106	USED_OIL_RECYCLER_ACTIVITY	N	Text	1		Y*	Y	HB070, HB073, HB075, HB077, HB081, HB083, HB085, HB092, HB102, HB110, HB112, HB120, HB190
29	107	USED_OIL_MARKET_BURNER	N	Text	1		Y*	N	HB072, HB073
30	108	USED_OIL_SPEC_MARKETER	N	Text	1		Y*	N	HB074, HB075
31	109	USED_OIL_BURNER	N	Text	1		Y*	N	HB076, HB077, HB081, HB083, HB085

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   At least one mode of transportation is required if TRANSPORTER_INDICATOR = **'C'** or **' X' and**   TRANSPORTER_RCRA_STATUS = **'R'**.
***   Required if TRANSPORTER_INDICATOR = **'C'** or **'X'**, else may be **blank**
****   Required if TSD_INDICATOR = '**X'**, else may be **blank**.

FLAT FILE ID# - HB (continued)

Table Name: HACTIVITY	IOR Type:	Variable
Description: Handler Activity Data	Decision Field:	SOURCE_TYPE
	Record Length:	407

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
32	110	USED_OIL_UTILITY_BOILER	N	Text	1		Y*	N	HB080, HB081
33	111	USED_OIL_INDUSTRIAL_BOILER	N	Text	1		Y*	N	HB082, HB083
34	112	USED_OIL_INDUSTRIAL_FURNACE	N	Text	1		Y*	N	HB084, HB085
35	113	USED_OIL_TRANSPORTER	N	Text	1		Y*	N	HB090, HB092
36	114	USED_OIL_PROCESSOR_REFINER	N	Text	1		Y*	N	HB100, HB102
37	115	USED_OIL_RCRA_STATUS	N	Text	1		Y*	Y**	HB110, HB112
38	116	USED_OIL_COLLECTION_SITE	N	Text	1		N	N	HB120
39	117	USED_OIL_STATE_STATUS	N	Text	1		N	N	--
40	118	USED_OIL_STATE_DESC	N	Text	12		N	N	--
41	130	HWFUEL_ACTIVITY	N	Text	1		Y*	Y	HB130, HB133, HB135, HB137, HB141, HB143, HB145, HB150, HB152, HB190

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required if USED_OIL_RECYCLER_IND = ' **X'**.
***   Skip two spaces. This will be deleted in a future translator guide version. The column USED_OIL_RCRA_DESC has been deleted from RCRAInfo.

FLAT FILE ID# - HB (continued)

Table Name: HACTIVITY **Description: Handler Activity Data**					IOR Type: Variable Decision Field: SOURCE_TYPE Record Length: 407				
No.	**Pos.**	**Field Name**	**Key**	**Type**	**Size**	**Sort**	**Nat. Req.**	**Sys. Req.**	**Edit #**
42	131	HWFUEL_MARKET_BURNER	N	Text	1		Y*	N	HB132, HB133
43	132	HWFUEL_OTHER_MARKETER	N	Text	1		Y*	N	HB134, HB135
44	133	HWFUEL_BURNER	N	Text	1		Y*	N	HB136, HB137, HB141, HB143, HB145
45	134	HWFUEL_UTILITY_BOILER	N	Text	1		Y*	N	HB140, HB141
46	135	HWFUEL_INDUSTRIAL_BOILER	N	Text	1		Y*	N	HB142, HB143
47	136	HWFUEL_INDUSTRIAL_FURNACE	N	Text	1		Y*	N	HB144, HB145
48	137	HWFUEL_RCRA_STATUS	N	Text	1		Y*	Y**	HB150, HB152
49	138	HWFUEL_STATE_STATUS	N	Text	1		N	N	–
50	139	HWFUEL_STATE_DESC	N	Text	12		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required if HWFUEL_ACTIVITY = **'B'**, '**D'**,'**E'**, or '**X'**.
***   Skip two spaces. This will be deleted in a future translator guide version. The column HWFUEL_RCRA_DESC has been deleted from RCRAInfo.

FLAT FILE ID# - HB (continued)

Table Name: HACTIVITY	IOR Type: Variable
Description: Handler Activity Data	Decision Field: SOURCE_TYPE
	Record Length: 407

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
51	151	UNIVERSAL_ACTIVITY	N	Text	1		N	N	HB160, HB162, HB164, HB190
52	152	UNIVERSAL_RCRA_STATUS	N	Text	1		N	N	HB162, HB164
53	153	UNIVERSAL_STATE_STATUS	N	Text	1		N	N	--
54	154	UNITVERSAL_STATE_DESC	N	Text	12		N	N	--
55	166	UNDERGROUND_INJECTION_ACTIVITY	N	Text	1		Y*	N	HB170, HB190
56	167	RECYCLER_ACTIVITY	N	Text	1		N	N	HB180, HB190
57	168	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Skip two spaces. This will be deleted in a future translator guide version. The column UNIVERSAL_RCRA_DESC has been deleted from RCRAInfo.

FLAT FILE ID# - HC

Table Name: HHAZARDOUS_WASTE_STREAM Description: Hazardous Waste Stream Data					IOR Type: Variable Decision Field: SOURCE_TYPE Record Length: 304					
No.	**Pos.**	**Field Name**	**Key**	**Type**	**Size**	**Sort**	**Nat. Req.**	**Sys. Req.**	**Edit #**	
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052	
2	2	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*)	Y	Text	12	1	N	Y	HC012, HC014, HC018, HC020	
3	14	ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*)	Y	Text	2	2	N	Y	HC000, HC012, HC014, HC018, HC020	
4	16	SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*)	Y	Text	1	3	N	Y	HC010, HC012, HC014, HC018, HC020	
5	17	WASTE_STREAM_SEQUENCE	Y	Numeric	4	4	N	Y	HC016, HC018, HC020	
6	21	WASTE_UNIT_OF_MEASURE	N	Text	1		N	Y**	HC030, HC032, HC034, HC038	
7	22	WASTE_STREAM_AMOUNT	N	Decimal - 9 whole positions, 5 decimal positions. INCLUDE DECIMAL POINT.	15		N	N	HC030, HC032, HC036, HC038	

FLAT FILE ID# - HC (continued)

Table Name: HHAZARDOUS_WASTE_STREAM Hazardous Waste Stream Data			Description:		IOR Type: Variable Decision Field: SOURCE_TYPE Record Length: 304						
8	37	WASTE_STREAM_DESC		N	Text	20		N	N	--
9	57	WASTE_STREAM_DATE		N	CCYYMMDD	8		N	N	HC050
10	65	NOTES		N	Text	240		N	N	--

*   Required for transaction processing.
**   Required if Waste Stream Amount is provided.

FLAT FILE ID# - HD

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
Table Name: HHAZARDOUS_WASTE_CODE   Description: IOR Type: Variable Hazardous Waste Code Data   Decision Field: SOURCE_TYPE   Record Length: 26
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052
2	2	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSOURCEFK_HH*)	Y	Text	12	1	N	Y	HD012, HD016, HD020, HD022
3	14	ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH*)	Y	Text	2	2	N	Y	HD000, HD012, HD016, HD020, HD022
4	16	SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSOURCESOURC*)	Y	Text	1	3	N	Y	HD010, HD012, HD016, HD020, HD022
5	17	WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*)	Y	Numeric	4	4	N	Y	HD014, HD016, HD020, HD022
6	21	OWNER (*FK_LU_HAND_WASTOWNER*)	Y	Text	2		N	Y	HD018, HD019, HD020, HD022
7	23	WASTE_CODE (*FK_LU_HAND_WASTWASTE_CODE*)	Y	Text	4		N	Y	HD019, HD020, HD022

*   Required for transaction processing.

FLAT FILE ID# - HE

<table>
<tr><td colspan="2">Table Name: HHAZARDOUS_WASTE_PROCESS
Hazardous Waste Associated Process Data</td><td>Description:</td><td colspan="6">IOR Type: Variable
Decision Field: SOURCE_TYPE
Record Length: 25</td></tr>
<tr><th>No.</th><th>Pos.</th><th>Field Name</th><th>Key</th><th>Type</th><th>Size</th><th>Sort</th><th>Nat. Req.</th><th>Sys. Req.</th><th>Edit #</th></tr>
<tr><td>1</td><td>1</td><td>TRANS_CODE</td><td>N</td><td>Text</td><td>1</td><td></td><td>N/A</td><td>Y*</td><td>CL050, CL052</td></tr>
<tr><td>2</td><td>2</td><td>HANDLER_ID (<i>FK_HHAZARDOUS_WFK_HSOURCEFK_HH</i>)</td><td>Y</td><td>Text</td><td>12</td><td>1</td><td>N</td><td>Y</td><td>HE012, HE016, HE020, HE022</td></tr>
<tr><td>3</td><td>14</td><td>ACTIVITY_LOCATION (<i>FK0HHAZARDOUS_WFK_HSOURCEFK_HH</i>)</td><td>Y</td><td>Text</td><td>2</td><td>2</td><td>N</td><td>Y</td><td>HE000, HE012, HE016, HE020, HE022</td></tr>
<tr><td>4</td><td>16</td><td>SOURCE_TYPE (<i>FK_HHAZARDOUS_WFK_HSOURCESOURC</i>)</td><td>Y</td><td>Text</td><td>1</td><td>3</td><td>N</td><td>Y</td><td>HE010, HE012, HE016, HE020, HE022</td></tr>
<tr><td>5</td><td>17</td><td>WASTE_STREAM_SEQUENCE (<i>FK_HHAZARDOUS_WWASTE_STREAM_SE</i>)</td><td>Y</td><td>Numeric</td><td>4</td><td>4</td><td>N</td><td>Y</td><td>HE014, HE016, HE020, HE022</td></tr>
<tr><td>6</td><td>21</td><td>OWNER (<i>FK_LU_PROCESS_COWNER</i>)</td><td>Y</td><td>Text</td><td>2</td><td></td><td>N</td><td>Y</td><td>HE018, HE019, HE020, HE022</td></tr>
<tr><td>7</td><td>23</td><td>PROCESS_CODE (<i>FK_LU_PROCESS_CPROCESS_CODE</i>)</td><td>Y</td><td>Text</td><td>3</td><td>5</td><td>N</td><td>Y</td><td>HE019, HE020, HE022</td></tr>
</table>

*   Required for transaction processing
**   Skip two spaces. This will be deleted in a future translator guide version. This foreign key created in error. There is no Unit of Measure lookup for waste associated processes.
***   Skip one space. This will be deleted in a future translator guide version. This foreign key was created in error. There is no Unit of Measure lookup for wastte associated processes.

FLAT FILE ID# - P1

Table Name:	PSERIES		IOR Type:	Fixed				
			Length:	296				
Description:	Permitting Series Data							

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	P1010, P1012, P1016, P1018
3	14	SERIES_SEQ	Y	Integer	3	2	Y*	Y	P1014, P1016, P1018
4	17	SERIES_NAME	N	Text	12		Y*	Y	P1020
5	29	EPAOWNER (*FK_LU_STAFF_EPAOWNER*)	N	Text	2		N	N	P1030, P1031
6	31	EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*)	N	Text	5		N	N	P1031
7	36	STATEOWNER (*FK_LU_STAFF_STATEOWNER*)	N	Text	2		N	N	P1040, P1041
8	38	STATEPERSON_ID (*FK_LU_STAFF_STATEPERSON_ID*)	N	Text	5		N	N	P1041
9	43	EPAATTYOWNER (*FK_LU_STAFF_EPAATTYOWNER*)	N	Text	2		N	N	P1050, P1051
10	45	EPAATTYPERSON_ID (*FK_LU_STAFF_EPAATTYPERSON_ID*)	N	Text	5		N	N	P1051
11	50	STATEATTYOWNER (*FK_LU_STAFF_STATEATTYOWNER*)	N	Text	2		N	N	P1060, P1061
12	52	STATEATTYPERSON_ID (*FK_LU_STAFF_STATEATTYPERSON_ID*)	N	Text	5		N	N	P1061
13	57	NOTES	N	Text	240		N	N	--

*　　　　　Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**　　　　Required for transaction processing.

FLAT FILE ID# - P2

Table Name: PEVENT **Description:** Permitting Event Data					**IOR Type:** Variable **IOR Decision Field:** Event Responsible Agency **Record Length:** 314				
No.	**Pos.**	**Field Name**	**Key**	**Type**	**Size**	**Sort**	**Nat. Req.**	**Sys. Req.**	**Edit #**
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_PSERIESFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	P2012, P2014, P2018, P2032, P2034
3	14	SERIES_SEQ (*FK_PSERIESSERIES_SEQ*)	Y	Integer	3	2	Y*	Y	P2016, P2032, P2034
4	17	OWNER (*FK_LU_PERMIT_EVOWNER*)	Y	Text	2		Y*	Y	P2020, P2021, P2022, P2023, P2032, P2034
5	19	PERMIT_EVENT_CODE (*FK_LU_PERMIT_EVPERMIT_EVENT_CO*)	Y	Text	7	3	Y*	Y	P2023, P2032, P2034
6	26	ACTIVITY_LOCATION	Y	Text	2	4	Y*	Y	P2000, P2012, P2032, P2034
7	28	RESPONSIBLE_AGENCY	Y	Text	1	5	Y*	Y	P2010, P2012, P2032, P2034, P2050, P2051, P2052

FLAT FILE ID# - P2 (Continued)

Table Name:	PEVENT				IOR Type:	Variable		
Description:	Permitting Event Data				IOR Decision Field:	Event Responsible Agency		
					Record Length:	314		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
8	29	EVENT_SEQ	Y	Integer	3	6	Y*	Y	P2030, P2032, P2034
9	32	ACTUAL_DATE	N	CCYYMMDD	8		Y*	Y***	P2040
10	40	SCHEDULE_DATE_ORIG	N	CCYYMMDD	8		N	N	P2042
11	48	SCHEDULE_DATE_NEW	N	CCYYMMDD	8		N	N	P2044
12	56	OWNER (*FK_LU_STAFFOWNER*)	N	Text	2		N	N	P2050, P2051, P2052, P2053
13	58	PERSON_IDENTIFIER (*FK_LU_STAFFPERSO_IDENTIFIER*)	N	Text	5		N	N	P2053
14	63	SUB_ORGOWNER	N	Text	2		N	N	
15	65	SUBORGANIZATION	N	Text	10		N	N	
16	75	NOTES	N	Text	240		N	N	--

*	Extracted for public access if EXTRACT_FLAG in HBASIC (H0) = **'X' and** PERMIT_EVENT_CODE is nationally required value.
**	Required for transaction processing.
***	If SCH_DATE_ORIG NE **Blank** or SCH_DATE_NEW NE **Blank**, ACTUAL_DATE may = **Blank.**

FLAT FILE ID# - P3

<table>
<tr><td colspan="5">Table Name: PUNIT
Description: Process Unit Group Data</td><td colspan="8">IOR Type: Fixed
Record Length: 275</td></tr>
<tr><td>No.</td><td>Pos.</td><td colspan="3">Field Name</td><td>Key</td><td>Type</td><td>Size</td><td>Sort</td><td>Nat. Req.</td><td>Sys. Req.</td><td>Edit #</td></tr>
<tr><td>1</td><td>1</td><td colspan="3">TRANS_CODE</td><td>N</td><td>Text</td><td>1</td><td></td><td>N/A</td><td>Y**</td><td>CL050, CL052</td></tr>
<tr><td>2</td><td>2</td><td colspan="3">HANDLER_ID (FK_HBASICHANDLER_ID)</td><td>Y</td><td>Text</td><td>12</td><td>1</td><td>Y*</td><td>Y</td><td>P3010, P3012, P3016, P3018</td></tr>
<tr><td>3</td><td>14</td><td colspan="3">UNIT_SEQ</td><td>Y</td><td>Integer</td><td>4</td><td>2</td><td>Y*</td><td>Y</td><td>P3014, P3016, P3018</td></tr>
<tr><td>4</td><td>18</td><td colspan="3">UNIT_NAME</td><td>N</td><td>Text</td><td>18</td><td></td><td>Y*</td><td>Y</td><td>P3020</td></tr>
<tr><td>5</td><td>36</td><td colspan="3">NOTES</td><td>N</td><td>Text</td><td>240</td><td></td><td>N</td><td>N</td><td>--</td></tr>
</table>

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'.**
**   Required for Transaction Processing

FLAT FILE ID# - P4

Table Name:	PUNIT_DETAIL		IOR Type:	Fixed						
Description:	Process Unit Group Detail Data		Record Length: 315							

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_PUNITFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	P4010, P4012, P4016, P4040, P4042
3	14	UNIT_SEQ (*FK_PUNITUNIT_SEQ*)	Y	Integer	4	2	Y*	Y	P4014, P4016, P4040, P4042
4	18	UNIT_DETAIL_SEQ	Y	Integer	3	3	Y*	Y	P4020, P4040, P4042
5	21	OWNER (*FK_LU_PROCESS_COWNER*)	N	Text	2		Y*	Y	P4050, P4051
6	23	PROCESS_CODE (*FK_LU_PROCESS_CPROCESS_CODE*)	N	Text	3		Y*	Y	P4051
7	26	OWNER (*FK_LU_PROCESS_CFK_LU_UOMOWNER*)	N	Text	2		Y*	Y	P4052
8	28	UNIT_OF_MEASURE_TYPE (*FK_LU_PROCESS_CFK_LU_UOM_TYPE*)	N	Text	1		Y*	Y	P4053
9	29	EFFECTIVE_DATE	N	CCYYMMDD	8		Y*	Y	P4060, P4062
10	37	CAPACITY	N	Decimal - 11 whole positions, 3 decimal positions. INCLUDE DECIMAL POINT.	15		Y*	Y	P4070, P4072
11	52	OWNER (*FK_LU_CAPACITY_OWNER*)	N	Text	2		N	N	P4080, P4081
12	54	CAPACITY_TYPE (*FK_LU_CAPACITY_CAPACITY_TYPE*)	N	Text	1		N	N	P4081
13	55	NUMBER_OF_UNITS	N	Integer	7		Y*	Y	P4090, P4091

FLAT FILE ID# - P4 (continued)

Table Name:	PUNIT_DETAIL		IOR Type:	Fixed						
Description:	Process Unit Group Detail Data		Record Length: 315							

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
14	62	OWNER (*FK_LU_COMMERCIAOWNER*)	N	Text	2		Y*	N	P4100, P4101
15	64	COMMERCIAL_STATUS_CODE (*FK_LU_COMMERCIACOMMERCIAL_STAT*)	N	Text	1		Y*	N	P4101
16	65	OWNER (*FK_LU_LEGALOPEOWNER*)	N	Text	2		Y*	Y	P4110, P4111
17	67	LEG_OP_STATUS_CODE (*FK_LU_LEGAL_OPELEG_OP_STATUS_C*)	N	Text	4		Y*	Y	P4111
18	71	OWNER (*FK_LU_SUBPROCESSOWNER*)	N	Text	2		N	N	P4120, P4121
19	73	SUBPROCESS_CODE (*FK_LU_SUBPROCESSSUBPROCESS_CODE*)	N	Text	3		N	N	P4121
20	76	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X' and** PROCESS_CODE is nationally required value.
**   Required for Transaction Processing

FLAT FILE ID# - P5

<table>
<tr><td colspan="6">Table Name: PLN_EVENT_UNIT
Description: Event to Process Unit Group Linking Data</td><td colspan="6">IOR Type: Variable
IOR Decision Field: Event Responsible Agency
Record Length: 38</td></tr>
<tr><th>No.</th><th>Pos.</th><th>Field Name</th><th>Key</th><th>Type</th><th>Size</th><th>Sort</th><th>Nat. Req.</th><th>Sys. Req.</th><th>Edit #</th></tr>
<tr><td>1</td><td>1</td><td>TRANS_CODE</td><td>N</td><td>Text</td><td>1</td><td></td><td>N/A</td><td>Y**</td><td>CL050, CL052</td></tr>
<tr><td>2</td><td>2</td><td>HANDLER_ID (FK_PEVENTFK_PSERIESFK_HBASICHA and FK_PUNITFK_HBASICHANDLER_ID)</td><td>Y</td><td>Text</td><td>12</td><td>1</td><td>Y*</td><td>Y</td><td>P5012, P5018, P5060, P5062, P5064, P5070, P5072</td></tr>
<tr><td>3</td><td>14</td><td>SERIES_SEQ (FKPEVENTFK_PSERIESSERIES_SEQ)</td><td>Y</td><td>Integer</td><td>3</td><td>2</td><td>Y*</td><td>Y</td><td>P5016, P5018, P5060, P5070, P5072</td></tr>
<tr><td>4</td><td>17</td><td>OWNER (FK_PEVENTFK_LU_PERMIT_EVOWNER)</td><td>Y</td><td>Text</td><td>2</td><td></td><td>Y*</td><td>Y</td><td>P5020, P5021, P5022, P5023, P5060, P5070, P5072</td></tr>
<tr><td>5</td><td>19</td><td>PERMIT_EVENT_CODE (FK_PEVENTFK_LU_PERMIT_EVPERMIT)</td><td>Y</td><td>Text</td><td>7</td><td>3</td><td>Y*</td><td>Y</td><td>P5023, P5060, P5070, P5072</td></tr>
<tr><td>6</td><td>26</td><td>ACTIVITY_LOCATION (FK_PEVENTACTIVITY_LOCATION)</td><td>Y</td><td>Text</td><td>2</td><td>4</td><td>Y*</td><td>Y</td><td>P5000, P5012, P5060, P5070, P5072</td></tr>
</table>

FLAT FILE ID# - P5 (Continued)

Table Name:	PLN_EVENT_UNIT			IOR Type:	Variable					
Description:	Event to Process Unit Group Linking Data			IOR Decision Field:	Event Responsible Agency					
				Record Length:	38					
7	28	RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_AGENCY*)		Y	Text	1	5	Y*	Y	P5010, P5012, P5060, P5070, P5072
8	29	EVENT_SEQ (*FK_PEVENTEVENT_SEQ*)		Y	Integer	3	6	Y*	Y	P5030, P5060, P5070, P5072
9	32	UNIT_SEQ (*FK_PUNITUNIT_SEQ*)		Y	Integer	4	7	Y*	Y	P5040, P5062, P5064, P5070, P5072
10	36	UNIT_DETAIL_SEQ_ (*FK_PUNIT_DETAILUNIT_DETAIL_SEQ*)		N	Integer	3	8	Y*	Y	P5050, P5064, P5070, P5072

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X' and** associated PERMIT_EVENT_CODE in PEVENT (P2) is natioanlly required value **and** associated PROCESS_CODE in PUNIT_DETAIL (P4) is nationally required value.
**   Required for Transaction Processing

FLAT FILE ID# - A1

Table Name:	AAREA		IOR Type:	Fixed
Description:	Corrective Action Area Data		Record Length:	317

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	A1010, A1012, A1016, A1018
3	14	AREA_SEQ	Y	Integer	4	2	Y*	Y	A1014, A1016, A1018
4	18	AREA_FACILITYWIDE_IND	N	Text	1		Y*	Y	A1030
5	19	AREA_NAME	N	Text	40		Y*	Y	A1020, A1040
6	59	AIR_RELEASE_IND	N	Text	1		Y*	N	A1050
7	60	GROUNDWATER_RELEASE_IND	N	Text	1		Y*	N	A1052
8	61	SOIL_RELEASE_IND	N	Text	1		Y*	N	A1054
9	62	SURFACE_WATER_RELEASE_IND	N	Text	1		Y*	N	A1056
10	63	REGULATED_UNIT_IND	N	Text	1		N	N	A1060
11	64	EPAOWNER (*FK_LU_STAFF_EPAOWNER*)	N	Text	2		N	N	A1070, A1071
12	66	EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*)	N	Text	5		N	N	A1071
13	71	STATEOWNER (*FK_LU_STAFF_STATEOWNER*)	N	Text	2		N	N	A1080, A1081
14	73	STATEPERSON_ID (*FK_LU_STAFF_STATEPERSON_ID*)	N	Text	5		N	N	A1081
15	78	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) = **'X'**.
**   Required for transaction processing.

FLAT FILE ID# - A2

Table Name:	ACA_AUTHORITY					IOR Type:	Variable		
Description:	Corrective Action Authority Data					IOR Decision Field:	Authority Responsible Agency		
						Record Length:	313		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	A2012, A2014, A2020, A2022
3	14	OWNER (*FK_LU_AUTHORITYOWNER*)	Y	Text	2		Y*	Y	A2016, A2020, A2022
4	16	AUTHORITY_TYPE (*FK_LU_AUTHORITYAUTHORITY_TYPE*)	Y	Text	1	2	Y*	Y	A2017, A2020, A2022
5	17	ACTIVITY_LOCATION	Y	Text	2	3	Y*	Y	A2000, A2012, A2020, A2022
6	19	RESPONSIBLE_AGENCY	Y	Text	1	4	Y*	Y	A2010, A2012, A2020, A2022
7	20	EFFECTIVE_DATE	Y	CCYYMMDD	8	5	Y*	Y	A2018, A2020, A2022
8	28	ISSUE_DATE	N	CCYYMMDD	8		Y*	N	A2030
9	36	REVOKE_DATE	N	CCYYMMDD	8		Y*	N	A2040
10	44	REPOSITORY	N	Text	1		N	N	A2050
11	45	OWNER (*FK_LU_LEAD_PROGOWNER*)	N	Text	2		Y*	N	A2060, A2061
12	47	LEAD_PROGRAM (*FK_LU_LEAD_PROGLEAD_PROGRAM*)	N	Text	1		Y*	N	A2061

FLAT FILE ID# - A2 (Continued)

Table Name:	ACA_AUTHORITY			IOR Type:	Variable		
Description:	Corrective Action Authority Data			IOR Decision Field:	Authority Responsible Agency		
				Record Length:	313		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
13	48	OWNER (*FK_LU_SUBORGANIOWNER*)	N	Text	2		N	N	A2070, A2071, A2072, A2073
14	50	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGINIZATION*)	N	Text	10		N	N	A2073
15	60	OWNER (*FK_LU_STAFFOWNER*)	N	Text	2		N	N	A2080, A2081, A2082, A2083
16	62	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text	5		N	N	A2083
17	67	OWNER (*FK_LU_STAFF_ATTYOWNER*)	N	Text	2		N	N	A2090, A2091, A2092, A2093
18	69	ATTYPERSON_ID (*FK_LU_STAFF_ATTYPERSON_ID*)	N	Text	5		N	N	A2093
19	74	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** CA_EVENT_CODE is nationally required value.
** Required for transaction processing.

FLAT FILE ID# - A3

Table Name:	ASTATUTE_CITATION				IOR Type:		Variable	
Description:	Corrective Action Statute/Citation Data				IOR Decision Field:		Authority Responsible Agency	
					Record Length:		270	

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_ACA_AUTHORITFK_HBASICHANDLE*)	Y	Text	12	1	Y*	Y	A3012, A3018, A3030, A3032
3	14	OWNER (*FK_ACA_AUTHORITFK_LU_AUTHOWNER*)	Y	Text	2		Y*	Y	A3014, A3015, A3018, A3030, A3032
4	16	AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AUTHTYPE*)	Y	Text	1	2	Y*	Y	A3015, A3018, A3030, A3032
5	17	ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_LOCATI*)	Y	Text	2	3	Y*	Y	A3000, A3012, A3018, A3030, A3032
6	19	RESPONSIBLE_AGENCY (for authority) (*FK_ACA_AUTHORITRESPONSIBLE_AGE*)	Y	Text	1	4	Y*	Y	A3010, A3012, A3018, A3030, A3032
7	20	EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIVE_DATE*)	Y	CCYYMMDD	8	5	Y*	Y	A3016, A3018, A3030, A3032
8	28	OWNER (*FK_LU_STATUTORYOWNER*)	Y	Text	2		Y*	Y	A3020, A3021, A3030, A3032

FLAT FILE ID# - A3 (Continued)

Table Name:	ASTATUTE_CITATION		IOR Type:	Variable					
Description:	Corrective Action Statute/Citation Data		IOR Decision Field:	Authority Responsible Agency					
			Record Length:	270					

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
9	30	STATUTORY_CITATION (*FK_LU_STATUTORYSTATUTORY_CITAT*)	Y	Text	1		Y*	Y	A3021, A3030, A3032
10	31	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** CA_EVENT_CODE is nationally required value.
**   Required for transaction processing.

FLAT FILE ID# - A4

Table Name:	AEVENT		IOR Type:	Variable		
Description:	Corrective Action Event Data		IOR Decision Field:	Event Responsible Agency		
			Record Length:	311		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	A4012, A4014, A4022, A4024
3	14	OWNER (*FK_LU_CA_EVENTOWNER*)	Y	Text	2		Y*	Y	A4016, A4017, A4018, A4019, A4022, A4024
4	16	CA_EVENT_CODE (*FK_LU_CA_EVENTCA_EVENT_CODE*)	Y	Text	7	2	Y*	Y	A4019, A4022, A4024
5	23	ACTIVITY_LOCATION	Y	Text	2	3	Y*	Y	A4000, A4012, A4022, A4024
6	25	RESPONSIBLE_AGENCY (for Event)	Y	Text	1	4	Y*	Y	A4010, A4012, A4022, A4024
7	26	EVENT_SEQ	Y	Integer	3	5	Y*	Y	A4020, A4022, A4024
8	29	ACTUAL_DATE	N	CCYYMMDD	8		Y*	Y***	A4040, A4042
9	37	SCHEDULE_DATE_ORIG	N	CCYYMMDD	8		N	N	A4040, A4044
10	45	SCHEDULE_DATE_NEW	N	CCYYMMDD	8		N	N	A4040, A4046
11	53	OWNER (*FK_LU_SUBORGANIOWNER*)	N	Text	2		N	N	A4050, A4051, A4052

FLAT FILE ID# - A4 (Continued)

					IOR Type:		Variable				
Table Name:		AEVENT			IOR Decision Field:		Event Responsible Agency				
Description:		Corrective Action Event Data			Record Length:		311				

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
12	55	RESPONSIBLE_SUB_ORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION*)	N	Text	10		N	N	
13	65	OWNER (*FK_LU_STAFFOWNER*)	N	Text	2		N	N	A4060, A4061, A4062, A4064
14	67	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text	5		N	N	A4064
15	72	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** CA_EVENT_CODE is nationally required value.
**   Required for transaction processing.
***   If SCH_DATE_ORIG NE **Blank** OR SCH_DATE_NEW NE **Blank**, ACTUAL_DATE may = **Blank.**

FLAT FILE ID# - A5

Table Name:	ALN_AREA_EVENT	IOR Type:	Variable
Description:	Event to CA Area Linking Data	IOR Decision Field:	Event Responsible Agency
		Record Length:	32

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_AAREAFK_HBASICHANDLER_ID* and *FK_AEVENTFK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	A5012, A5040, A5042, A5050, A5052
3	14	AREA_SEQ (*FK_AAREAAREA_SEQ*)	Y	Integer	4	2	Y*	Y	A5030, A5042
4	18	OWNER (*FK_AEVENTFK_LU_CAEVENTOWNER*)	Y	Text	2		Y*	Y	A5014, A5015, A5016, A5018, A5040, A5050, A5052
5	20	CA_EVENT_CODE (*FK_AEVENTFK_LU_CA_EVENTCA_EVEN*)	Y	Text	7	3	Y*	Y	A5018, A5040, A5050, A5052
6	27	ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION*)	Y	Text	2	4	Y*	Y	A5000, A5040, A5050, A5052
7	29	RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*)	Y	Text	1	5	Y*	Y	A5010, A5040, A5050, A5052
8	30	EVENT_SEQ (*FK_AEVENTEVENT_SEQ*)	Y	Integer	3	6	Y*	Y	A5020, A5040, A5050, A5052

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** linked (asssociated) CA_EVENT_CODE in AEVENT (A4) is nationally required value.
**   Required for Transaction Processing.

FLAT FILE ID# - A6

Table Name:	ALN_EVENT_AUTHORITY				IOR Type:	Variable		
Description:	Event to CA Authority Linking Data				IOR Decision Field: Event Responsible Agency			
					Record Length: 40			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_AEVENTFK_HBASICHANDLER_ID* and *FK_ACA_AUTHORITFK_HBASICHANDLE)*	Y	Text	12	1	Y*	Y	A6012, A6032, A6034, A6040, A6042, A6050, A6052
3	14	OWNER (*FK_AEVENTFK_LU_CA_EVENTOWNER*)	Y	Text	2		Y*	Y	A6014, A6015, A6016, A6018, A6040, A6042, A6050, A6052
4	16	CA_EVENT_CODE (*FK_AEVENTFK_LU_CA_EVENTCA_EVEN*)	Y	Text	7	3	Y*	Y	A6018, A6040, A6042, A6050, A6052
5	23	ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION* and *FK_ACA_AUTHORITACTIVIT_LOCATI)*	Y	Text	2	4	Y*	Y	A6000, A6012, A6032, A6040, A6042, A6050, A6052
6	25	RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*)	Y	Text	1	5	Y*	Y	A6010, A6012, A6032, A6040, A6042, A6050, A6052

FLAT FILE ID# - A6 (Continued)

Table Name:		ALN_EVENT_AUTHORITY		IOR Type:		Variable				
Description:		Event to CA Authority Linking Data		IOR Decision Field:		Event Responsible Agency				
				Record Length:		40				
7	26	EVENT_SEQ (*FK_AEVENTEVENT_SEQ*)	Y	Integer	3	6	Y*	Y	A6020, A6040
8	29	OWNER (*FK_ACA_AUTHORITFK_LU_AUTHOWNER*)	Y	Text	2		Y*	Y	A6036, A6037
9	31	AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AUTHTYPE*)	Y	Text	1	2	Y*	Y	A6037
11	32	RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSIBLE_AGE*)	Y	Text	1	7	Y*	Y	A6030
12	33	EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIVE_DATE*)	Y	CCYYMMDD	8	8	Y*	Y	A6038, A6042, A6050, A6052

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** linked (asssociated) CA_EVENT_CODE in AEVENT (A4) is nationally required value.
**   Required for Transaction Processing.

FLAT FILE ID# - C1

<table>
<tr>
<td colspan="4">Table Name: CEVALUATION
Description: CM&E Evaluation Data</td>
<td colspan="5">IOR Type: Variable
IOR Decision Field: Evaluation Agency
Record Length: 296</td>
</tr>
<tr>
<th>No.</th>
<th>Pos.</th>
<th>Field Name</th>
<th>Key</th>
<th>Type</th>
<th>Size</th>
<th>Sort</th>
<th>Nat. Req.</th>
<th>Sys. Req.</th>
<th>Edit #</th>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>TRANS_CODE</td>
<td>N</td>
<td>Text</td>
<td>1</td>
<td></td>
<td>N/A</td>
<td>Y**</td>
<td>CL050,
CL052</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>HANDLER_ID (*FK_HBASICHANDLER_ID*)</td>
<td>Y</td>
<td>Text</td>
<td>12</td>
<td>1</td>
<td>Y*</td>
<td>Y</td>
<td>C1002,
C1004,
C1006,
C1008</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>ACTIVITY_LOCATION</td>
<td>Y</td>
<td>Text</td>
<td>2</td>
<td>2</td>
<td>Y*</td>
<td>Y</td>
<td>C1000,
C1004,
C1006,
C1008</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>EVALUATION_DATE</td>
<td>Y</td>
<td>CCYYMMDD</td>
<td>8</td>
<td>3</td>
<td>Y*</td>
<td>Y</td>
<td>C1010,
C1004,
C1008</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>EVALUATION_SEQ</td>
<td>Y</td>
<td>Text</td>
<td>3</td>
<td>4</td>
<td>Y*</td>
<td>Y</td>
<td>C1004,
C1008,
C1014</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>AGENCY</td>
<td>Y</td>
<td>Text</td>
<td>1</td>
<td>5</td>
<td>Y*</td>
<td>Y</td>
<td>C1004,
C1008,
C1012</td>
</tr>
<tr>
<td>7</td>
<td>28</td>
<td>OWNER (*FK_LU_EVALUATIOOWNER*)</td>
<td>N</td>
<td>Text</td>
<td>2</td>
<td></td>
<td>Y*</td>
<td>Y</td>
<td>C1026,
C1028</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>EVALUATION_TYPE
(*FK_LU_EVALUATIOEVALUATION_TYPE*)</td>
<td>N</td>
<td>Text</td>
<td>3</td>
<td></td>
<td>Y*</td>
<td>Y</td>
<td>C1028</td>
</tr>
<tr>
<td>9</td>
<td>33</td>
<td>FOUND_VIOLATION</td>
<td>N</td>
<td>Text</td>
<td>1</td>
<td></td>
<td>Y*</td>
<td>N</td>
<td>C1016,
C1018,
C1020</td>
</tr>
<tr>
<td>10</td>
<td>34</td>
<td>OWNER (*FK_LU_EVALUATIOREASONOWNER*)</td>
<td>N</td>
<td>Text</td>
<td>2</td>
<td></td>
<td>N</td>
<td>N</td>
<td>C1022,
C1024</td>
</tr>
<tr>
<td>11</td>
<td>36</td>
<td>REASON_TYPE (*FK_LU_EVALUATIONREASON_TYPE*)</td>
<td>N</td>
<td>Text</td>
<td>2</td>
<td></td>
<td>N</td>
<td>N</td>
<td>C1024</td>
</tr>
<tr>
<td>12</td>
<td>38</td>
<td>OWNER (*FK_LU_SUBORGANIOWNER*)</td>
<td>N</td>
<td>Text</td>
<td>2</td>
<td></td>
<td>N</td>
<td>N</td>
<td>C1030,
C1032</td>
</tr>
</table>

FLAT FILE ID# - C1 (Continued)

Table Name: Description:		CEVALUATION CM&E Evaluation Data	IOR Type: IOR Decision Field: Record Length:		Variable Evaluation Agency 296				
13	40	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION*)	N	Text	10		N	N	C1032
14	50	OWNER (*FK_LU_STAFFOWNER*)	N	Text	2		N	N	C1034, C1036
15	52	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text	5		N	N	C1036
16	57	NOTES	N	Text	240		N	N	--

* Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** EVALUATION_TYPE is nationally required value.
** Required for transaction processing.

FLAT FILE ID# - C2

Table Name:	CEVALUATION_AREA	IOR Type:	Variable
Description:	CM&E Evaluation Coverage Area Data	IOR Decision Field:	Evaluation Agency
		Record Length:	274

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052
2	2	HANDLER_ID (*FK_CEVALUATIONFK_HBASICHANDLER*)	Y	Text	12	1	N	Y	C2002, C2004, C2006, C2008
3	14	ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_LOCATIO*)	Y	Text	2	2	N	Y	C2000, C2004, C2006, C2008
4	16	EVALUATION_DATE (*FK_CEVALUATIONEVALUATION_DATE*)	Y	CCYYMMDD	8	3	N	Y	C2004, C2008
5	24	EVALUATION_SEQ (*FK_CEVALUATIONEVALUATION_SEQ*)	Y	Text	3	4	N	Y	C2004, C2008
6	27	AGENCY (*FK_CEVALUATIONAGENCY*)	Y	Text	1	5	N	Y	C2004, C2008
7	28	OWNER (*FK_LU_EVALUATIOOWNER*)	Y	Text	2		N	Y	C2012, C2014
8	30	AREA_TYPE (*FK_LU_EVALUATIOAREA_TYPE*)	Y	Text	3	6	N	Y	C2004, C2008, C2014
9	33	EVALUATED	N	Text	2		N	Y	C2010
10	35	NOTES	N	Text	240		N	N	--

*   Required for transaction processing.

FLAT FILE ID# - C3

Table Name:	CVIOLATION			IOR Type:	Variable		
Description:	CM&E Violation Data			IOR Decision Field:	Violation Responsible Agency		
				Record Length:	344		

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Y	Text	12	1	Y*	Y	C3002, C3004, C3006, C3008
3	14	ACTIVITY_LOCATION	Y	Text	2	2	Y*	Y	C3000, C3004, C3006, C3008
4	16	DETERMINED_BY_AGENCY	Y	Text	1	3	Y*	Y	C3004, C3008, C3010
5	17	VIOLATION_SEQ	Y	Text	4	4	Y*	Y	C3004, C3008, C3012
6	21	DETERMINED_DATE	N	CCYYMMDD	8		Y*	Y	C3014, C3016, C3018
7	29	RESPONSIBLE_AGENCY	N	Text	1		Y*	Y	C3020
8	30	OWNER (*FK_LU_EVALUATIOOWNER*)	N	Text	2		Y*	Y	C3028, C3030
9	32	AREA_TYPE (*FK_LU_EVALUATIOAREA_TYPE*)	N	Text	3		Y*	Y	C3030
10	35	CITATION	N	Text	30		Y*, ***	N	C3026
11	65	ACTUAL_RTC_DATE	N	CCYYMMDD	8		Y*	N	C3022, C3024
12	73	OWNER (*FK_LU_CLASSOWNER*)	N	Text	2		Y*	N	C3032, C3034
13	75	CLASS_TYPE (*FK_LU_CLASSCLASS_TYPE*)	N	Text	1		Y*	N	C3034

FLAT FILE ID# - C3 (continued)

Table Name:	CVIOLATION		IOR Type:	Variable	
Description:	CM&E Violation Data		IOR Decision Field:	Violation Responsible Agency	
			Record Length:	344	

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
14	76	OWNER (*FK_LU_PRIORITYOWNER*)	N	Text	2		Y*	N	C3038, C3040
15	78	PRIORITY_TYPE (*FK_LU_PRIORITYPRIORITY_TYPE*)	N	Text	1		Y*	N	C3040
16	79	OWNER (*FK_LU_RTC_QUALIOWNER*)	N	Text	2		Y*	N	C3042, C3044
17	81	RTC_QUALIFIER (*FK_LU_RTC_QUALIRTC_QUALIFIER_T*)	N	Text	1		Y*	N	C3044, C3046
18	82	OWNER (*FK_LU_CITATION_OWNER*)	N	Text	2		Y*	N	C3048, C3050
19	84	CITATION_TYPE (*FK_LU_CITATION_CITATION_TYPE*)	N	Text	2		N	N	C3050, C3052
20	86	OWNER (*FK_LU_SUBORGANIOWNER*)	N	Text	2		N	N	C3054, C3056
21	88	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION*)	N	Text	10		N	N	C3056
22	98	OWNER (*FK_LU_STAFFOWNER*)	N	Text	2		N	N	C3058, C3060
23	100	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text	5		N	N	C3060
24	105	NOTES	N	Text	240		N	N	--

* Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** AREA_TYPE is nationally required value and not enforcement sensitive. CLASS_TYPE = 'P' is enforcement sensitive if it has not been addressed by an enforcement action or if the only enforcement action addressing it is enforcement sensative. (Enforcement sensitive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)

** Required for transaction processing.

*** Considered nationally required if VIOLATION_CITATION = **266.102**, **266.103**, **266.104**, **266.105**, **266.106**, **266.107**, **266.111**, or **266.112**.

FLAT FILE ID# - C4

Table Name:	CLN_EVALUATION_VIOLATION		IOR Type:	Variable	
Description:	CM&E Evalation to Violations Linking Data		IOR Decision Field:	Evaluation Agency	
			Record Length:	32	

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_CEVALUATIONFK_HBASICHANDLER* and *FK_CVIOLATIONFK_HBASICHANDLER*)	Y	Text	12	1	Y*	Y	C4002, C4004, C4006, C4008, C4012, C4014
3	14	ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_LOCATIO* and *FKCVIOLATIONACTIVITY_LOCATIO*)	Y	Text	2	2	Y*	Y	C4000, C4004, C4006, C4008, C4012, C4014
4	16	EVALUATION_DATE (*FK_CEVALUATIONEVALUATION_DATE*)	Y	CCYYMMDD	8	3	Y*	Y	C4004, C4008, C4012
5	24	EVALUATION_SEQ (*FK_CEVALUATIONEVALUATION_SEQ*)	Y	Text	3	4	Y*	Y	C4004, C4008, C4012
6	27	AGENCY (*FK_CEVALUATIONAGENCY*)	Y	Text	1	5	Y*	Y	C4004, C4008, C4012
7	28	DETERMINED_BY_AGENCY (*FK_CVIOLATIONDETERMINED_BY_AGE*)	Y	Text	1	6	Y*	Y	C4004, C4008, C4014
8	29	VIOLATION_SEQ (*FK_CVIOLATIONVIOLATION_SEQ*)	Y	Text	4	7	Y*	Y	C4004, C4008, C4014

* Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** EVALUATION_TYPE is nationally required value **and** AREA_TYPE is nationally required value and not enforcement sensitive.   CLASS_TYPE = 'P' is enforcement sensitive if it has not been addressed by an enforcement action or if the only enforcement action addressing it is enforcement sensative. (Enforcement sensitive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)

** Required for transaction processing.

FLAT FILE ID# - C5

Table Name:	CENFORCEMENT		IOR Type:	Variable
Description:	CM&E Enforcement Data		IOR Decision Field:	Enforcement Agency
			Record Length:	311

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys.Re q.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID *(FK_HBASICHANDLER_ID)*	Y	Text	12	1	Y*	Y	C5002, C5004, C5006, C5008
3	14	ACTIVITY_LOCATION	Y	Text	2	2	Y*	Y	C5000, C5004, C5006, C5008
4	16	ENFORCEMENT_DATE	Y	CCYYMMDD	8	3	Y*	Y	C5004, C5008, C5012, C5014
5	24	ENFORCEMENT_SEQ	Y	Text	3	4	Y*	Y	C5004, C5008, C5010
6	27	AGENCY	Y	Text	1	5	Y*	Y	C5004, C5008, C5016
7	28	OWNER *(FK_LU_ENFORCEMEOWNER)*	N	Text	2		Y*	Y	C5018, C5020
8	30	ENFORCEMENT_TYPE *(FK_LU_ENFORCEMEENFORECEMENT_TYP)*	N	Text	3		Y*	Y	C5020
9	33	DOCKET	N	Text	15		N	N	--
10	48	OWNER *(FK_LU_SUBORGANIOWNER)*	N	Text	2		N	N	C5022, C5024
11	50	SUBORGANIZATION *(FK_LU_SUBORGANISUBORGANIZATION)*	N	Text	10		N	N	C5024
12	60	OWNER *(FK_LU_STAFFOWNER)*	N	Text	2		N	N	C5026, C5028

FLAT FILE ID# - C5 (Continued)

Table Name:		CENFORCEMENT			IOR Type:		Variable				
Description:		CM&E Enforcement Data			IOR Decision Field:		Enforcement Agency				
					Record Length:		311				
13	62	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text	5		N	N	C5028
14	67	ATTORNEY	N	Text	5		N	N	--
15	72	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** ENFORCEMENT_TYPE is nationally required value and not enforcement sensitive. (Enforcement sensitive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)

**   Required for transaction processing.

FLAT FILE ID# - C6

Table Name:	CMEDIA			IOR Type:	Variable				
Description:	CM&E Multimedia Information			IOR Decision Field:	Enforcement Agency				
				Record Length:	272				

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*)	Y	Text	12	1	Y*	Y	C6002, C6004, C6006, C6008
3	14	ACTIVITY_LOCATION (*FK_CENFORCEMENTACTIVITY_LOCATI*)	Y	Text	2	2	Y*	Y	C6000, C6004, C6006, C6008
4	16	ENFORCEMENT_DATE (*FK_CENFORCEMENTENFORCEMENT_DAT*)	Y	CCYYMMDD	8	3	Y*	Y	C6004, C6008
5	24	ENFORCEMENT_SEQ (*FK_CENFORCEMENTENFORCEMENT_SEQ*)	Y	Text	3	4	Y*	Y	C6004, C6008
6	27	AGENCY (*FK_CENFORCEMENTAGENCY*)	Y	Text	1	5	Y*	Y	C6004, C6008
7	28	OWNER (*FK_LU_MEDIA_TYPOWNER*)	Y	Text	2		Y*	Y	C6010, C6012
8	30	MEDIA_TYPE (*FK_LU_MEDIA_TYPMEDIA_TYPE*)	Y	Text	3	7	Y*	Y	C6004, C6008, C6012
9	33	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** ENFORCEMENT_TYPE is nationally required   value and not enforcement sensitive **and** MEDIA_TYPE is nationally required value. (Enforcement sensitive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)

**   Required for transaction processing.

FLAT FILE ID# - C7

Table Name:	CSEP	IOR Type:	Variable
Description:	CM&E Supplemental Environmental Project	IOR Decision Field:	Enforcement Agency
		Record Length:	310

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*)	Y	Text	12	1	Y*	Y	C7002, C7004, C7006, C7008
3	14	ACTIVITY_LOCATION (*FK_CENFORCEMENTACTIVITY_LOCATI*)	Y	Text	2	2	Y*	Y	C7000, C7004, C7006, C7008
4	16	ENFORCEMENT_DATE (*FK_CENFORCEMENTENFORCEMENT_DAT*)	Y	CCYYMMDD	8	3	Y*	Y	C7004, C7008
5	24	ENFORCEMENT_SEQ (*FK_CENFORCEMENTENFORCEMENT_SEQ*)	Y	Text	3	4	Y*	Y	C7004, C7008
6	27	AGENCY (*FK_CENFORCEMENTAGENCY*)	Y	Text	1	5	Y*	Y	C7004, C7008
7	28	SEP_SEQ	Y	Text	2	6	Y*	Y	C7004, C7008, C7010
8	30	OWNER (*FK_LU_SEP_TYPEOWNER*)	N	Text	2		Y*	Y	C7020, C7022
9	32	SEP_TYPE (*FK_LU_SEP_TYPESEPTYPE*)	N	Text	3		Y*	Y	C7022
10	35	EXPENDITURE_AMOUNT	N	Decimal - 10 whole positions, 2 decimal positions. INCLUDE DECIMAL POINT.	13		N	N	--
11	48	SCHED_DATE	N	CCYYMMDD	8		N	N	C7011, C7012, C7014
12	56	ACTUAL_DATE	N	CCYYMMDD	8		N	N	C7011, C7016, C7018

FLAT FILE ID# - C7 (Continued)

Table Name:	CSEP		IOR Type:		Variable					
Description:	CM&E Supplemental Environmental Project		IOR Decision Field: Record Length:		Enforcement Agency 310					
13	64	OWNER (*FK_LU_STAFFOWNER*)	N	Text		2		N	N	C7024, C7026
14	66	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	N	Text		5		N	N	C7026
15	71	NOTES	N	Text		240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** ENFORCEMENT_TYPE is nationally required value and not enforcement sensitive **and** SEP_TYPE is nationally required value. (Enforcement sensistive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)

**   Required for transaction processing.

FLAT FILE ID# - C8

Table Name:	CPENALTY		IOR Type:	Variable				
Description:	CM&E Penalty Data		IOR Decision Field:	Enforcement Agency				
			Record Length:	285				

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*)	Y	Text	12	1	Y*	Y	C8002, C8004, C8006, C8008
3	14	ACTIVITY_LOCATION (*FK_CENFORCEMENTACTIVITY_LOCATI*)	Y	Text	2	2	Y*	Y	C8000, C8004, C8006, C8008
4	16	ENFORCEMENT_DATE (*FK_CENFORCEMENTENFORCEMENT_DAT*)	Y	CCYYMMDD	8	3	Y*	Y	C8004, C8008
5	24	ENFORCEMENT_SEQ (*FK_CENFORCEMENTENFORCEMENT_SEQ*)	Y	Text	3	4	Y*	Y	C8004, C8008
6	27	AGENCY (*FK_CENFORCEMENTAGENCY*)	Y	Text	1	5	Y*	Y	C8004, C8008
7	28	OWNER (*FK_LU_PENALTY_TOWNER*)	Y	Text	2				C8012, C8014
8	30	PENALTY_TYPE (*FK_LU_PENALTY_TPENALTY_TYPE*)	Y	Text	3	6	Y*	Y	C8004, C8008, C8014, C8016, C8018
9	33	AMOUNT	N	Decimal - 10 whole positions, 2 decimal positions. INCLUDE DECIMAL POINT.	13		Y*	Y	C8010
10	46	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HBASIC (H0) table = **'X' and** ENFORCEMENT_TYPE is nationally required value and not enforcement sensitive.
         (Enforcement sensistive actions are ENFORCEMENT_TYPE = '400-499' and '810' or '700-799' without a final monetary penalty (PENALTY_TYPE = 'FA').)
**   Required for transaction processing.

FLAT FILE ID# - C9

Table Name:	CPAYMENT					IOR Type:	Variable			
Description:	CM&E Payment Data					IOR Decision Field:	Enforcement Agency			
						Record Length:	316			

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y*	CL050, CL052
2	2	HANDLER_ID (*FK_CPENALTYFK_CENFORCEMENTFK_H*)	Y	Text	12	1	N	Y	C9002, C9004, C9006, C9008
3	14	ACTIVITY_LOCATION (*FK_CPENALTYFK_CENFORCEMENTACTI*)	Y	Text	2	2	N	Y	C9000, C9004, C9006, C9008
4	16	ENFORCEMENT_DATE (*FK_CPENALTYFK_CENFORCEMENTDATE*)	Y	CCYYMMDD	8	3	N	Y	C9004, C9008
5	24	ENFORCEMENT_SEQ (*FK_CENFORCEMENTENFORCEMENT_SEQ*)	Y	Text	3	4	N	Y	C9004, C9008
6	27	AGENCY (*FKCPENALTYFK_CENFORCEMENTAGEN*)	Y	Text	1	5	N	Y	C9004, C9008
7	28	OWNER (*FK_CPENALTYFK_LU_PENALTY_TOWNE*)	Y	Text	2		N	N	C9024, C9026
8	30	PENALTY_TYPE (*FK_CPENALTYFK_LU_PENALTY_TPENA*)	Y	Text	3	6	N	Y	C9004, C9008, C9026
9	33	PAYMENT_SEQ	Y	Text	2	7	N	Y	C9004, C9008, C9010
10	35	SCHED_DATE	N	CCYYMMDD	8		N	Y**	C9012, C9014
11	43	SCHED_AMOUNT	N	Decimal - 10 whole positions, 2 decimal positions. INCLUDE DECIMAL POINT.	13		N	Y***	C9016

FLAT FILE ID# - C9 (Continued)

Table Name:	CPAYMENT			IOR Type:	Variable					
Description:	CM&E Payment Data			IOR Decision Field: Enforcement Agency						
				Record Length: 316						
12	56	PAID_DATE	N	CCYYMMDD	8		N	N	C9018, C9020
13	64	PAID_AMOUNT	N	Decimal - 10 whole positions, 2 decimal positions. INCLUDE DECIMAL POINT.	13		N	N	C9022
14	77	NOTES	N	Text	240		N	N	---

*   Required for transaction processing.
**   If SCHED_AMOUNT is entered, SCHED_DATE is required.
***   If SCHED_DATE is entered, SCHED_AMOUNT is required.

FLAT FILE ID# - CA

Table Name:	CLN_COMPLIANCE_SCHEDULE		IOR Type:	Variable
Description:	CM&E Violation to Enforcement Action Linking Data		IOR Decision Field:	Enforcement Agency
			Record Length:	280

No.	Pos.	Field Name	Key	Type	Size	Sort	Nat. Req.	Sys. Req.	Edit #
1	1	TRANS_CODE	N	Text	1		N/A	Y**	CL050, CL052
2	2	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE* and *FK_CVIOLATIONFK_HBASICHANDLER*)	Y	Text	12	1	Y*	Y	CA002, CA004, CA006, CA008, CA040, CA042
3	14	ACTIVITY_LOCATION (*FK_CENFORCEMENTACTIVITY_LOCATI* and *FK_CVIOLATIONACTIVITY_LOCATION)*	Y	Text	2	2	Y*	Y	CA000, CA004, CA006, CA008, CA040, CA042
4	16	ENFORCEMENT_DATE (*FK_CENFORCEMENTENFORCEMENT_DAT*)	Y	CCYYMMDD	8	3	Y*	Y	CA004, CA008, CA040
5	24	ENFORCEMENT_SEQ (*FK_CENFORCEMENTENFORCEMENT_SEQ*)	Y	Text	3	4	Y*	Y	CA004, CA008, CA040
6	27	AGENCY (*FK_CENFORCEMENTAGENCY*)	Y	Text	1	5	Y*		CA004, CA008, CA040
7	28	DETERMINED_BY_AGENCY (*FK_CVIOLATIONDETERMINED_BY_AGE*)	Y	Text	1	6	Y*	Y	CA004, CA008, CA042
8	29	VIOLATION_SEQ (*FK_CVIOLATIONVIOLATION_SEQ*)	Y	Text	4	7	Y*	Y	CA004, CA008, CA042
9	33	COMPLIANCE_SCHED_DATE	N	CCYYMMDD	8		Y	N	CA010, CA012
10	41	NOTES	N	Text	240		N	N	--

*   Extracted for public access if EXTRACT_FLAG flag in HHANDLER (H1) table = **'X' and** ENFORCEMENT_TYPE is nationally required value **and** AREA_TYPE is nationally required value.
**   Required for transaction processing.

APPENDIX B: TRANSLATOR DATA ASSESSMENT EDITS

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
Control File (CL) Edits				
CL010	FLATFILE_NAME	Invalid State Code in FLATFILE_NAME	First two characters of FLATFILE_NAME NOT valid State Postal Code. If **True**, set ABORT_FLAG = **ON** Continue.	; PRE-LOAD ERRORS All edits on control file columns are *pre-load* edits. This means that if an error(s) is detected, the translator load process will not continue.
CL012	FLATFILE_NAME	Invalid Flat File Identifier in FLATFILE_NAME	Characters 3 and 4 of FLATFILE_NAME NE 'CL', 'H0 - H8', 'HA - HE', 'P1 - P5', 'A1 - A6', 'C1 - C9', or 'CA'. If **True**, set ABORT_FLAG = **ON** Continue.	
CL014	FLATFILE_NAME	Invalid Julian Date in FLATFILE_NAME	Julian Date NE numeric value derived from DATE_CREATED. If **True**, set ABORT_FLAG = **ON** Continue.	
CL016	FLATFILE_NAME	FLATFILE_NAME previously processed.	FLATFILE_NAME NOT unique for a given Julian Date. If **True**, set ABORT_FLAG = **ON** Continue.	Sequence indicator is used to differentiate multiple flat file load submissions on the same day.
CL020	SOFTWARE_VERSION	Invalid SOFTWARE_VERSION	SOFTWARE_VERSION NE first digit of RCRAInfo software version. If **True**, set ABORT_FLAG = **ON** Continue.	
CL030	DATE_CREATED	Invalid DATE_CREATED	Not valid date OR when converted to Julian Date is NE Julian Date supplied in FLATFILE_NAME. If **True**, set ABORT_FLAG = **ON**. Continue.	
CL040	RECORD_COUNT	RECORD_COUNT NOT= actual record count in flat file.	RECORD_COUNT NE actual count of records submitted for FLATFILE_NAME Flat File Identifier If **True**, set ABORT_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CL050	FULL_REPLACE_IND, TRANS_CODE	FULL_REPLACE_IND=Y, TRANS_CODE not blank.	FULL_REPLACE_IND in Control (CL) File = **Y** AND TRANS_CODE in any flat file record for a module NE **Blank**. If **True**, set ABORT_FLAG = **ON**, Continue.	
CL051	FULL_REPLACE_IND	FULL_REPLACE_IND must = Y, N or blank	FULL_REPLACE_IND in Control(CL) File NE **Y**, **N** or **blank** If **True**, set ABORT FLAG = **ON**, Continue.	
CL052	FULL_REPLACE_IND, TRANS_CODE	TRANS_CODE NOT= C or D	″ FULL_REPLACE_IND in Control (CL) File = **Blank** AND TRANS_CODE in any flat file record for a module NE **C** OR **D**. If **True**, set ABORT_FLAG = ON. ″ CONTROL FILE just validated is control file for last flat file submitted. If **True** AND ABORT_FLAG = **ON**, Abort translator load. If **True** AND ABORT_FLAG NE **ON**, continue to other flat file edits. If **False**, validate CONTROL FILE for next flat file in series.	
HBASIC (H0) Edits				
H0000	HANDLER_ID	Invalid State Code for &handler_id	First two characters of HANDLER_ID NOT valid State Postal Code. If **True**, reject record.	First two digits of a Handler ID must always be a valid state prefix.
H0002	HANDLER_ID	State prefix for HANDLER_ID NOT = state being processed.	First 2 characters of HANDLER_ID NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	First two characters of HANDLER_ID must be same as state being processed.
H0010	HANDLER_ID	Not IOR for HBASIC block for &activity_location.	Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HBASIC. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HBASIC data can have one owner per state prefix.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H0012	HANDLER_ID	&handler_id exists in HBASIC; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID exists in HBASIC table. If **True**, reject record.	This checks for duplicate flat file records.
H0014	HANDLER_ID	Unowned data linked to HBASIC; Cannot delete &handler_id.	TRANS_CODE = **D** AND row matching HANDLER_ID exists in HBASIC table AND an unowned data row (s) is linked to any row in any table associated with that HANDLER_ID. If **True**, reject record.	
H0016	HANDLER_ID	**WARNING**: HANDLER_ID &handler_id fails algorithm check.	(TRANS_CODE = **C** OR **Blank**) AND row matching HANDLER_ID NOT in HBASIC table AND HANDLER_ID fails check-digit algorithm. If **True**, display warning message, set **Warning Flag** for later use.	Allows implementers to add State ID numbers which don't pass federal check-digit algorithm to the database.
H0018	HANDLER_ID	&handler_id missing in HBASIC; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID NOT found in HBASIC table. If **True**, reject record.	
H0020	HANDLER_NAME	HANDLER_NAME cannot be blank	(TRANS_CODE = **C** OR **Blank**) AND HANDLER_NAME = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other fields in record. At END, if ERROR_FLAG = ON, reject record.
H0030	EXTRACT_FLAG	EXTRACT_FLAG must be blank or 'X'	(TRANS_CODE = **C** OR **Blank**) AND EXTRACT_FLAG NE **Blank.** OR **'X'** If **True**, set ERROR_FLAG = **ON**, Continue. If ERROR_FLAG = **ON**, reject record.	
–	FACILITY_IDENTIFIER	–	–	No edit on this data column.
–	NOTES	–	–	No edit on this data column.
HHANDLER (H1) Edits				
H1000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H1010	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	Not IOR for HHANDLER block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H1 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HHANDLER. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HHANDLER data can have one owner per ACTIVITY_LOCATION. HHANDLER data is fixed IOR within the context of a Region/State pair for a particular ACTIVITY_LOCATION.
H1012	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process HHANDLER record.	Row matching HANDLER_ID NOT found in HBASIC or TR_HBASIC table. If **True,** reject record.	Referential integrity check.
H1014	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION	HHANDLER row exists; Cannot add.	TRANS_CODE = **Blank**. AND row matching HANDLER_ID AND ACTIVITY_LOCATION exists in HHANDLER table. If **True**, reject record.	This checks for duplicate flat file records.
H1016	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE = **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HBASIC. Determine if AUTHORIZED_AGENCY = owner group for Translator. If **True**, reject record.	
H1018	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION	HHANDLER row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HHANDLER table. If **True**, reject record.	
H1020	OWNER (*FK_LU_COUNTYOWNER*)	COUNTY CODE Owner must = HQ.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER NOT = **HQ**). If **True**, set ERROR_FLAG ON, Continue to LOCATION_STREET1 edit (H1030).	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H1021	COUNTY_CODE (*FK_LU_COUNTYCOUNTY_CODE),* OWNER (*FK_LU_COUNTYOWNER*)	COUNTY CODE not in LU table	(TRANS_CODE = **C** OR **Blank**) AND COUNTY_CODE for OWNER **NOT** found in LU_COUNTY table. If **True**, set ERROR_FLAG ON, Continue.	
–	LOCATION_STREET_NO	–	–	No edit on this data column..
H1030	LOCATION_STREET1	LOCATION_STREET1 cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND LOCATION_STREET1 = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other fields in record. At END, If ERROR_FLAG = ON, reject record.
–	LOCATION_STREET2	–	–	No edit on this data column.
H1032	LOCATION_CITY	LOCATION_CITY cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND LOCATION_CITY = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H1034	LOCATION_STATE	LOCATION_STATE must match 1st 2 chars of HANDLER_ID.	(TRANS_CODE = **C** OR **Blank**) AND LOCATION_STATE NOT = 1st 2 characters of HANDLER_ID (*FK_HBASIC_HANDLER_ID*). If **True**, set ERROR_FLAG = **ON**, Continue.	
H1036	LOCATION_ZIP	LOCATION_ZIP cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND LOCATION_ZIP = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
–	MAIL_STREET_NO	–	–	No edit on this data column.
H1040	MAIL_STREET1	MAIL_STREET1 cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND (NON_NOTIFIER NE **X** OR **E**) AND MAIL_STREET1 = **Blank.** If **True**, set ERROR_FLAG = **ON**, Continue.	
–	MAIL_STREET2	–	–	No edit on this data column.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H1042	MAIL_CITY	MAIL_CITY cannot be blank	(TRANS_CODE = **C** OR **Blank**) AND (NON_NOTIFIER NE **X** OR **E**) AND MAIL_CITY = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H1044	MAIL_STATE_CODE	Invalid MAIL_STATE_CODE.	(TRANS_CODE = **C** OR **Blank**) AND (NON_NOTIFIER NE **X** OR **E**) AND MAIL_STATE_CODE NOT valid FIPS state or foreign country code. If **True**, set ERROR_FLAG = **ON**, Continue.	Mailing addresses can be foreign addresses.
H1046	MAIL_ZIP	MAIL_ZIP cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND (NON_NOTIFIER NE **X** OR **E**) AND MAIL_ZIP = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H1050	LAND_TYPE	LAND_TYPE must = C,D,I,F,S,M,O,P, or blank.	(TRANS_CODE = **C** OR **Blank**) AND LAND_TYPE NE **C** OR **D** OR **I** OR **F** OR **S** OR **M** OR **O** OR **P** OR **Blank**. If **True**, set ERROR_FLAG = **ON,** Continue.	
–	PREVIOUS_ID	–	–	No edit on this data column.
H1060	NON_NOTIFIER	NON_NOTIFIER must = E,O,X, or blank.	(TRANS_CODE = **C** OR **Blank**) AND NON_NOTIFIER NE **E** OR **O** OR **X** OR **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H1070	ACCESSIBILITY	ACCESSIBILITY must = B,F,C,L, or blank.	(TRANS_CODE = **C** OR **Blank**) AND ACCESSIBILITY NE **B** OR **F** OR **C** OR **L** OR **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H1080	TSD_DATE	Invalid TSD_DATE	(TRANS_CODE = **C** OR **Blank**) AND TSD_DATE NOT valid date or **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H1090	OFF_SITE_RECEIPT	OFF_SITE_RECEIPT must = A,R,N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND OFF_SITE_RECEIPT NE **A** OR **R** OR **N** OR **Blank**. If **True**, set ERROR_FLAG = **ON**., 　　　Continue.	
H1100	SECOND_ID	HANDLER_ID in H0 failed algorithm check. on add SECOND_ID must = HANDLER_ID.	(TRANS_CODE = **C** OR **Blank**) AND (Warning Flag from new handler add transaction as part of H0 processing is set ON) AND SECOND_ID NE HANDLER_ID). If **True,** set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	If HANDLER_ID does not pass algorithm check, SECOND_ID must be same as HANDLER_ID to indicate that ID was not added to the database in error.
–	STATE_DISTRICT	–	–	No edit on this data column.
–	RIVER_BASIN	–	–	No edit on this data column.
–	NOTES	–	–	No edit on this data column.
HPREVIOUS_NAME (H2) Edits				
H2000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H2010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Not IOR for HHANDLER block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H2 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HHANDLER. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HPREVIOUS_NAME data can have one owner per ACTIVITY_LOCATION. HPREVIOUS_NAME is part of a fixed IOR block with HHANDLER.
H2012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol (HHANDLER); Cannot process HPREVIOUS_NAME record.	Row matching HANDLER_ID AND ACTIVITY_LOCATION NOT found in HHANDLER table If **True,** reject record.	Referential integrity check.
H2014	RECEIVE_DATE	Invalid RECEIVE_DATE	RECEIVE_DATE not valid date. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H2016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), RECEIVE_DATE	HPREVIOUS_NAME row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND RECEIVE_DATE exists in HPREVIOUS_NAME table. If **True**, reject record.	
H2018	HANDLER_ID (*FK_HHANDLERFK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), RECEIVE_DATE	HPREVIOUS_NAME row missing; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND RECEIVE_DATE **NOT** found in HPREVIOUS_NAME table. If **True**, reject record.	
H2020	PREVIOUS_HANDLER_NAME	PREVIOUS_HANDLER_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND PREVIOUS_HANDLER_NAME = **Blank**. If **True**, reject record.	
–	NOTES	–	–	No edit on this data column.
HLATITUDE_LONGITUDE (H3) Edits				
H3000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H3010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Not IOR for HHANDLER block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H3 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HHANDLER. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HLATITUDE_LONGITUDE data can have one owner per ACTIVITY_LOCATION. HLATITUDE_LONGITUDE is part of a fixed IOR block with HHANDLER.
H3012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HLATITUDE_LONGITUDE record.	Row matching HANDLER_ID AND ACTIVITY_LOCATION NOT found in HHANDLER table If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H3014	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	HLATITUDE_LONGITUDE row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION exists in HLATITUDE_LONGITUDE table. If **True**, reject record.	
H3016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	HLATITUDE_LONGITUDE row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HLATITUDE_LONGITUDE table. If **True**, reject record.	
H3020	LATITUDE_LONGITUDE_SOURCE	LATITUDE_LONGITUDE_SOURCE must = Z, A, V, or I.	(TRANS_CODE = **C** OR **Blank**) AND LATITUDE_LONGITUDE_SOURCE NE **Z** OR **A** OR **V** OR **I**. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other fields in record. At END, If ERROR_FLAG = ON, reject record.
H3030	LATITUDE_MEASURE	Invalid LATITUDE_MEASURE	(TRANS_CODE = **C** OR **Blank**) AND LATITUDE_MEASURE does **NOT** meet the following criteria: Positions 1-2: 00-90 (degrees) Positions 3-4: 00-59 (minutes) Positions 5-6: 00-59 (seconds) Position 7: 0-9 (tenths of second). If **True**, set ERROR_FLAG = **ON**, Continue.	
H3040	LONGITUDE_MEASURE	Invalid LONGITUDE_MEASURE	(TRANS_CODE = **C** OR **Blank**) AND LONGITUDE_MEASURE does **NOT** meet the following criteria: Positions 1-3: 000-180 (degrees) Positions 4-5: 00-59 (minutes) Postitions 6-7: 00-59 (seconds) Position 8 : 0-9 (tenths of second) If **True**, set ERROR_FLAG = **ON**. If ERROR_FLAG = **ON**, reject record.	
–	NOTES	–	–	No edit on this data column.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HENV_PRIORITY_RANK (H4) Edits				
H4000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H4010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Not IOR for HHANDLER block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H4 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HHANDLER. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HENV_PRIORITY_RANK data can have one owner per ACTIVITY_LOCATION. HENV_PRIORITY_RANK is part of a fixed IOR block with HHANDLER.
H4012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HENV_PRIORITY_RANK record.	Row matching HANDLER_ID AND ACTIVITY_LOCATION NOT found in HHANDLER table If **True,** reject record.	Referential integrity check.
H4014	EPR_DATE	Invalid EPR_DATE.	Not valid date. If **True**, reject record.	
H4016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), EPR_DATE	HENV_PRIORITY_RANK row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND EPR_DATE exists in HENV_PRIORITY_RANK table. If **True**, reject record.	
H4018	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), EPR_DATE.	HENV_PRIORITY_RANK row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND EPR_DATE **NOT** found in HENV_PRIORITY_RANK table. If **True**, reject record.	
H4020	EPR_STATUS	EPR_STATUS must = H, M, or L.	(TRANS_CODE = **C** OR **Blank**) AND EPR_STATUS NE **H** OR **M** OR **L**. If **True**, reject record.**.**	
–	NOTES	–	–	No edit on this data column.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HCONTACT (H5) Edits				
H5000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H5010	OWNER (*FK_LU_CONTACTOWNER*)	CONTACT_TYPE owner must = HQ or translator's Region or State code.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER NOT = **HQ** OR Region or State code matching that of translator submitting job). If **True**, reject record.	The identity of the translator is determined from information received via variable provided as part of load submittal process.
H5011	CONTACT_TYPE (*FK_LU_CONTACTCONTACT_TYPE*), OWNER (*FK_LU_CONTACTOWNER*)	CONTACT_TYPE not in LU table for Owner &owner.	(TRANS_CODE = **C** OR **Blank**) AND CONTACT_TYPE for OWNER **NOT** found in LU_CONTACT table. If **True**, reject record.	
H5020	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OWNER (*FK_LU_CONTACTOWNER*),CONTACT_TYPE (*FK_LU_CONTACTCONTACT_TYPE*)	Not IOR for HCONTACT block for ACTIVITY_LOCATION &activity_location and CONTACT_TYPE &contact_type.	Match ACTIVITY_LOCATION in H5 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = HCONTACT. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching CONTACT_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HCONTACT is a variable IOR block. CONTACT_TYPE is the decision column. HCONTACT data can have one owner per ACTIVITY_LOCATION and CONTACT_TYPE.
H5022	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HCONTACT record.	Row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HHANDLER table If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H5024	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OWNER (*FK_LU_CONTACTOWNER*), CONTACT_TYPE (*FK_LU_CONTACTCONTACT_TYPE*)	HCONTACT row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OWNER AND CONTACT_TYPE exists in HCONTACT table. If **True**, reject record.	
H5026	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OWNER (*FK_LU_CONTACTOWNER*), CONTACT_TYPE (*FK_LU_CONTACTCONTACT_TYPE*)	HCONTACT_TYPE row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OWNER AND CONTACT_TYPE **NOT** found in HCONTACT table. If **True**, reject record.	
–	CONTACT_TITLE	–	--	No edit on this data column.
H5030	CONTACT_LAST_NAME	CONTACT_LAST_NAME cannot be blank..	(TRANS_CODE = **C** OR **Blank**) AND CONTACT_LAST_NAME = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other field in record. At END, If ERROR_FLAG = ON, reject record.
–	CONTACT_FIRST_NAME	–	–	No edit on this data column.
H5040	STREET1	STREET1 cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND STREET1 = **Blank.** If **True**, set ERROR_FLAG = **ON**, Continue.	
–	STREET2	–	–	No edit on this data column.
H5042	CITY	CITY cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND CITY = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H5044	STATE	STATE must be valid state postal or foreign country code.	(TRANS_CODE = **C** OR **Blank**) AND STATE = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H5046	ZIP	ZIP cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND ZIP = **Blank.** If **True**, set ERROR_FLAG = **ON**. If ERROR_FLAG = **ON**, reject record.	
–	PHONE	–	–	No edit on this data column.
–	NOTES	–	–	No edit on this data column.
HOWNER_OPERATOR (H6) Edits				
H6000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H6010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Not IOR for HOWNER_OPERATOR block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H6 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HOWNER_OPERATOR . Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HOWNER_OPERATOR data can have one owner per ACTIVITY_LOCATION. HOWNER_OPERATOR is a fixed IOR block.
H6012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HOWNER_OPERATOR	Row matching HANDLER_ID AND ACTIVITY_LOCATION NOT found in HHANDLER table If **True,** reject record.	Referential integrity check.
H6014	OWNER_OPERATOR_SEQ	OWNER_OPERATOR_SEQ not numeric.	OWNER_OPERATOR_SEQ **NOT** numeric. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H6016	HANDLER_ID (*FK_HHANDLERFK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OWNER_OPERATOR_SEQ	HOWNER_OPERATOR row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OWNER_OPERATOR_SEQ exists in HOWNER_OPERATOR table. If **True**, reject record.	
H6018	HANDLER_ID (*FK_HHANDLERFK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OWNER_OPERATOR_SEQ.	HOWNER_OPERATOR row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OWNER_OPERATOR_SEQ **NOT** found in HOWNER_OPERATOR table. If **True**, reject record.	
H6020	OWNER_OPERATOR_INDICATOR	OWNER_OPERATOR_INDICATOR must = CO, CP, PO, or PP.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER_OPERATOR_INDICATOR NE **CO** OR **CP** OR **PO** OR **PP**). If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other field in record. At END, If ERROR_FLAG = ON, reject record.
H6030	OWNER_OPERATOR_TYPE	OWNER_OPERATOR_TYPE must = F, S, P, I, C, M, D, or O.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER_OPERATOR_TYPE NE **F** OR **S** OR **P** OR **I** OR **C** OR **M** OR **D** OR **O**). If **True**, set ERROR_FLAG = **ON,** Continue.	
H6040	OWNER_OPERATOR_CHANGE_DATE	OWNER_OPERATOR_CHANGE_DATE must be blank when OWNER_OPERATOR_TYPE = CO or CP.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER_OPERATOR_TYPE = **CO** OR **CP**) AND OWNER_OPERATOR_CHANGE_DATE **NOT** = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H6042	OWNER_OPERATOR_CHANGE_DATE	OWNER_OPERATOR_CHANGE_DATE must be valid date when OWNER_OPERATOR_TYPE = PO or PP.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER_OPERATOR_TYPE = **PO** OR **PP**) AND OWNER_OPERATOR_CHANGE_DATE **NOT** valid date. If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H6050	OWNER_OPERATOR_NAME	OWNER_OPERATOR_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND OWNER_OPERATOR_NAME = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H6060	PHONE	PHONE cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND PHONE = **Blank**. If True, set ERROR_FLAG = **ON**, Continue.	
H6070	STREET1	STREET1 cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND STREET1 = **Blank.** If **True**, set ERROR_FLAG = **ON,** Continue.	
–	STREET2	–	–	No edit on this data column.
H6072	CITY	CITY cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND CITY = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H6074	STATE	STATE cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND STATE = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	
H6076	ZIP	ZIP cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND ZIP = **Blank.** If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	
–	NOTES	–	–	No edit on this data column.
HSIC (H7) Edits				
H7000	ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H7010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Not IOR for HSIC block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H7 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HSIC. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HSIC data can have one owner per ACTIVITY_LOCATION . HSIC is a fixed IOR block.
H7012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HSIC	Row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HHANDLER table If **True,** reject record.	Referential integrity check.
H7014	SIC_SEQUENCE	SIC_SEQUENCE not numeric.	SIC_SEQUENCE **NOT** numeric. If **True**, reject record..	
H7016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), SIC_SEQUENCE	HSIC row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SIC_SEQUENCE exists in HSIC table. If **True**, reject record.	
H7018	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), SIC_SEQUENCE.	HSIC row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SIC_SEQUENCE **NOT** found in HSIC table. If **True**, reject record.	
H7020	OWNER (*FK_LU_SICOWNER*)	SIC_CODE owner must = HQ.	(TRANS_CODE = **C** OR **Blank**) AND OWNER NE **HQ**. If **True**, set ERROR_FLAG = **ON**, Continue to PRIMARY_SIC_INDICATOR edit (H7030).	
H7021	SIC_CODE (*FK_LU_SICSICCODE*), OWNER (*FK_LU_SICOWNER)*	SIC_CODE not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND SIC_CODE for OWNER **NOT** found in LU_SIC table. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other field in record. At END, If ERROR_FLAG = ON, reject record.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H7030	PRIMARY_SIC_INDICATOR	PRIMARY_SIC_INDICATOR must = P or S.	(TRANS_CODE = **C** OR **Blank**) AND (PRIMARY_SIC_INDICATOR NE **P** OR **S**). If **True**, set ERROR_FLAG = **ON**, Continue.	
H7040	SIC_SOURCE	SIC_SOURCE must = R or D.	(TRANS_CODE = **C** OR **Blank**) AND (SIC_SOURCE NE **R** OR **D**). If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record	
HOTHER_PERMIT (H8) Edits				
H8000	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
H8010	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Not IOR for HOTHER_PERMIT block for ACTIVITY_LOCATION &activity_location.	Match ACTIVITY_LOCATION in H8 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = HOTHER_PERMIT. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	HOTHER_PERMIT data can have one owner per ACTIVITY_LOCATION. HOTHER_PERMIT is a fixed IOR block .
H8012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol (HHANDLER); Cannot process HOTHER_PERMIT.	Row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HHANDLER table If **True,** reject record.	Referential integrity check.
H8014	OTHER_PERMIT_NUMBER	OTHER_PERMIT_NUMBER cannot be blank.	OTHER_PERMIT_NUMBER = **Blank.** If **True**, reject record.	
H8016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*), OTHER_PERMIT_NUMBER	HOTHER_PERMIT row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OTHER_PERMIT_NUMBER exists in HOTHER_PERMIT table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
H8018	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), OTHER_PERMIT_NUMBER.	HOTHER_PERMIT row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND OTHER_PERMIT_NUMBER **NOT** found in HOTHER_PERMIT table. If **True**, reject record.	
H8020	OWNER (*FK_LU_OTHER_PEROWNER)*	OTHER_PERMIT_TYPE Owner must = US.	(TRANS_CODE = **C** OR **Blank**) AND OWNER NE **US**. If **True**, reject record..	
H8021	OTHER_PERMIT_TYPE (*FK_LU_OTHER_PEROTHER_PERMIT_TY)*, OWNER (*FK_LU_OTHER_PEROWNER)*	Invalid OTHER_PERMIT_TYPE	(TRANS_CODE = **C** OR **Blank**) AND OTHER_PERMIT_TYPE for OWNER **NOT** found in LU_OTHER_PERMIT_TYPE table. If **True**, reject record.	
–	OTHER_PERMIT_DESC	–	–	No edit on this data column.
–	NOTES	–	–	No edit on this data column.
HSOURCE (HA) Edits				
HA000	ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION)*	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
HA010	SOURCE_TYPE	SOURCE_TYPE must = N, A, S, E, or R. Cannot determine IOR.	SOURCE_TYPE NE **N** OR **A** OR **S** OR **E** OR **R**. If **True**, reject record.	
HA012	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*), SOURCE_TYPE	Not IOR for HSOURCE block for ACTIVITY_LOCATION &activity_location and SOURCE_TYPE &source_type.	Match ACTIVITY_LOCATION in HA record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HSOURCE**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching SOURCE_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HSOURCE is a variable IOR block with HHANDLER. The decision column for HSOURCE is SOURCE_TYPE. HSOURCE data can have one owner per ACTIVITY_LOCATION and SOURCE_TYPE.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HA014	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*)	Ref. Integrity Viol. (HHANDLER); Cannot process HSOURCE.	Row matching HANDLER_ID AND ACTIVITY_LOCATION **NOT** found in HHANDLER table If **True,** reject record.	Referential integrity check.
HA016	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*), SOURCE_TYPE	HSOURCE row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE exists in HSOURCE table. If **True**, reject record.	
HA018	HANDLER_ID (*FK_HHANDLERFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_HHANDLERACTIVITY_LOCATION*), SOURCE_TYPE	HSOURCE row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE **NOT** found in HSOURCE table. If **True**, reject record.	
–	NOTES	–	–	No edit on this data column.
HACTIVITY (HB) Edits				
HB000	ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
HB010	SOURCE_TYPE	SOURCE_TYPE must = N, A, S, E, or R. Cannot determine IOR.	SOURCE_TYPE NE **N** OR **A** OR **S** OR **E** OR **R**. If **True**, reject record.	
HB012	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*)	Not IOR for HSOURCE block for ACTIVITY_LOCATION &activity_location and SOURCE_TYPE &source_type.	Match ACTIVITY_LOCATION in HB record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HSOURCE**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching SOURCE_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HACTIVITY is part of a variable IOR block with HSOURCE. HACTIVITY data can have one owner per ACTIVITY_LOCATION and SOURCE_TYPE.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB014	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*)	Ref. Integrity Viol. (HSOURCE); Cannot process HACTIVITY.	Row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE **NOT** found in HSOURCE table. If **True,** reject record.	Referential integrity check.
HB016	ACTIVITY_SEQ	ACTIVITY_SEQ must be numeric.	ACTIVITY_SEQ NOT numeric. If **True**, reject record.	
HB018	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FKHSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), ACTIVITY_SEQ.	HACTIVITY row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND ACTIVITY_SEQ exists in HACTIVITY table. If **True**, reject record.	
HB020	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FKHSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), ACTIVITY_SEQ	HACTIVITY row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND ACTIVITY_SEQ **NOT** found in HACTIVITY table. If **True**, reject record.	
HB030	RECEIVE_DATE	Invalid RECEIVE_DATE.	Not valid date. If **True**, reject record.	
HB040	OWNER (*FK_LU_GENERATOROWNER*), GENERATOR_ACTIVITY (*FK_LU_GENERATORGENERATOR_ACTIV*)	GENERATOR ACTIVITY OWNER must = HQ or translator's Region or State code	(TRANS_CODE = **C** OR **Blank**) AND GENERATOR_ACTIVITY NE **Blank** AND (OWNER NE **HQ** OR Region or State code matching that of translator submitting job). If **True**, set ERROR_FLAG = **ON**, Continue to GENERATOR_RCRA_STATUS edit (HB042).	Information received via variable provided as part of load submittal process. If field fails edit, software applies remaining edits to other fields in record. At END, If ERROR_FLAG = ON, reject record.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB041	GENERATOR_ACTIVITY (*FK_LU_GENERATORGENERATOR_ACTIV*), OWNER (*FK_LU_GENERATOROWNER*)	GENERATOR_ACTIVITY not found in LU table.	(TRANS_CODE = **C** OR **Blank**) AND GENERATOR_ACTIVITY NE **Blank** AND GENERATOR_ACTIVITY for OWNER NOT found in LU_GENERATOR table. If **True**, set ERROR_FLAG = **ON**, Continue.	
HB042	GENERATOR_RCRA_STATUS	GENERATOR_RCRA_STATUS must = R, P, A, N or blank.	(TRANS_CODE = **C** OR **Blank**) AND (GENERATOR_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB044	GENERATOR_RCRA_DESC	GENERATOR_RCRA_DESC must = 1-10 or blank.	(TRANS_CODE = **C** OR **Blank**) AND (GENERATOR_RCRA_DESC NE **01** OR **02** OR **03** OR **04** OR **05** OR **06** OR **07** OR **08** OR **09** OR **10** or **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
--	GENERATOR_STATE_STATUS	–	--	No edit on this data column.
--	GENERATOR_STATE_DESC	–	--	No edit on this data column.
HB050	TRANSPORTER_ACTIVITY	TRANSPORTER_ACTIVITY must = C, S, X, N or blank	(TRANS_CODE = **C** OR **Blank**) AND (TRANSPORTER_ACTIVITY NE **C** OR **S** OR **X** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON,** Continue.	
HB051	AIR	AIR must = X or blank.	(TRANS_CODE = **C** or **Blank**) AND (AIR NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB052	RAIL	RAIL must = X or blank.	(TRANS_CODE = **C** or **Blank**) AND (RAIL NE **X** or **Blank**). If **True**, set ERROR_FLAG = **ON,** Continue.	
HB053	HIGHWAY	HIGHWAY must = X or blank.	(TRANS_CODE = **C** or **Blank**) AND (HIGHWAY NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB054	WATER	WATER must = X or blank	(TRANS_CODE = **C** OR **Blank**) AND (WATER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
--	OTHER_MODE	–	--	No valid value edit on this data column.
HB056	TRANSPORTER_RCRA_STATUS, TRANSPORTER_ACTIVITY	TRANSPORTER_RCRA_STATUS must = R, P, A, or N.	(TRANS_CODE = **C** OR **Blank**) AND (TRANSPORTER_ACTIVITY = **C** OR **X**) AND (TRANSPORTER_RCRA_STATUS NE **R** OR **P** OR **A** OR **N**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB057	TRANSPORTER_RCRA_STATUS, TRANSPORTER_ACTIVITY	TRANSPORTER_RCRA_STATUS must = R, P, A, N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (TRANSPORTER_ACTIVITY NE **C** OR **X**) AND (TRANSPORTER_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**.) If **True**, set ERROR_FLAG = **ON,** Continue.	
HB058	AIR, RAIL, HIGHWAY, WATER, OTHER_MODE, TRANSPORTER_RCRA_STATUS, TRANSPORTER_ACTIVITY	Method of transportation required-- TRANSPORTER_ACTIVITY = C or X and TRANSPORTER_RCRA_STATUS = R.	(TRANS_CODE = **C** OR **Blank**.) AND (TRANSPORTER_ACTIVITY = **C** OR **X**) AND (TRANSPORTER_RCRA_STATUS = **R**) AND (AIR = **Blank** AND RAIL = **Blank** AND HIGHWAY = **Blank** AND OTHER_MODE = **Blank**) If **True**, set ERROR_FLAG = **ON,** Continue.	At least one method of transportation is required for RCRA regulated transporters unless handler transports waste for self (TRANSPORTER_ACTIVITY = S).
HB059	TRANSPORTER_RCRA_DESC	TRANSPORTER_RCRA_DESC must = 1, 2, 3, 4, or blank..	(TRANS_CODE = **C** OR **Blank**) AND (TRANSPORTER_RCRA_DESC NE **1** OR **2** OR **3** OR **4** OR **Blank**) If True, set ERROR_FLAG = **ON**, Continue.	
--	TRANSPORTER_STATE_STATUS	–	–	No edit on this data column.
--	TRANSPORTER_STATE_DESC	–	–	No edit on this data column.
HB060	TSD_ACTIVITY	TSD_ACTIVITY must = X, N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (TSD_ACTIVITY NE **X** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB062	TSD_RCRA_STATUS, TSD_ACTIVITY	TSD_RCRA_STATUS must = R, P, A, or N.	(TRANS_CODE = **C** OR **Blank**) AND (TSD_ACTIVITY = **X**) AND (TSD_RCRA_STATUS NE **R** OR **P** OR **A** OR **N**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB064	TSD_RCRA_STATUS, TSD_ACTIVITY	TSD_RCRA_STATUS must = R, P, A, N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (TSD_ACTIVITY NE **X**) AND (TSD_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**.) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB066	TSD_RCRA_DESC	TSD_RCRA_DESC must = 1-7 or blank.	(TRANS_CODE = **C** OR **Blank**) AND (TSD_RCRA_DESC NE **1** OR **2** OR **3** OR **4** OR **5** OR **6** OR **7** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
--	TSD_STATE_STATUS	–	–	No edit on this data column.
--	TSD_STATE_DESC	–	–	No edit on this data column.
HB070	USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X, N, or blank..	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_RECYCLER_ACTIVITY NE **X** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB072	USED_OIL_MARKET_BURNER	USED_OIL_MARKET_BURNER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_MARKET_BURNER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB073	USED_OIL_MARKET_BURNER, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_MARKET_BURNER = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_MARKET_BURNER = **X** AND USED_OIL_RECYCLER_ACTIVITY NE **X**. If **True**, set ERROR_FLAG = **ON**, Continue.	
HB074	USED_OIL_SPEC_MARKETER	USED_OIL_SPEC_MARKETER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_SPEC_MARKETER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB075	USED_OIL_SPEC_MARKETER, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_SPEC_MARKETER = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_SPEC_MARKETER = **X** AND USED_OIL_RECYCLER_ACTIVITY NE **X**. If **True**, set ERROR_FLAG = **ON**, Continue.	
HB076	USED_OIL_BURNER	USED_OIL_BURNER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_BURNER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB077	USED_OIL_BURNER, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_BURNER = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_BURNER = **X** AND USED_OIL_RECYCLER_ACTIVITY NE **X**. If **True**, set ERROR_FLAG = **ON**, Continue.	
HB080	USED_OIL_UTILITY_BOILER	USED_OIL_UTILITY_BOILER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_UTILITY_BOILER NE **X** or **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB081	USED_OIL_UTILITY_BOILER, USED_OIL_RECYCLER_ACITIVITY, USED_OIL_BURNER.	USED_OIL_RECYCLER_ACTIVITY must = X and USED_OIL_BURNER must = X if USED_OIL_UTILITY_BOILER = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_UTILITY_BOILER = **X** AND (USED_OIL_RECYCLER_ACTIVITY NE **X** OR USED_OIL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB082	USED_OIL_INDUSTRIAL_BOILER	USED_OIL_INDUSTRIAL_BOILER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_INDUSTRIAL_BOILER NE **X** OR **Blank**.) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB083	USED_OIL_INDUSTRIAL_BOILER, USED_OIL_RECYCLER_ACITIVITY, USED_OIL_BURNER.	USED_OIL_RECYCLER_ACTIVITY must = X and USED_OIL_BURNER must = X if USED_OIL_INDUSTRIAL_BOILER = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_INDUSTRIAL_BOILER = **X** AND (USED_OIL_RECYCLER_ACTIVITY NE **X** OR USED_OIL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB084	USED_OIL_INDUSTRIAL_FURNACE	USED OIL_INDUSTRIAL_FURNACE must = X or blank	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_INDUSTRIAL_FURNACE NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB085	USED_OIL_INDUSTRIAL_FURNACE, USED_OIL_RECYCLER_ACTIVITY, USED_OIL_BURNER	USED_OIL_RECYCLER_ACTIVITY must = X and USED_OIL_BURNER must = X if USED_OIL_INDUSTRIAL_FURNACE = X.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_INDUSTRIAL_FURNACE = **X** AND (USED_OIL_RECYCLER_ACTIVITY NE **X** OR USED_OIL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB090	USED_OIL_TRANSPORTER	USED_OIL_TRANSPORTER must = T, B, F, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_TRANSPORTER NE **T** OR **B** OR **F** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB092	USED_OIL_TRANSPORTER, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_TRANSPORTER = T, B, or F.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_TRANSPORTER = **T** OR **B** OR **F** AND USED_OIL_RECYCLER_ACTIVITY NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB100	USED_OIL_PROCESSOR_REFINER	USED_OIL_PROCESSOR_REFINER must = P, B, R, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_PROCESSOR_REFINER NE **P** OR **B** OR **R** or **Blank**). If True, set ERROR_FLAG = **ON**, Continue.	
HB102	USED_OIL_PROCESSOR_REFINER, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_PROCESSOR_REFINER = P, B, or R.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_PROCESSOR_REFINER = **P** OR **B** OR **R** AND USED_OIL_RECYCLER_ACTIVITY NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB110	USED_OIL_RCRA_STATUS, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RCRA_STATUS must = R, P, A, or N.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_RECYCLER_ACTIVITY = **X**) AND (USED_OIL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N**) If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB112	USED_OIL_RCRA_STATUS, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RCRA_STATUS must = R, P, A, N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_RECYCLER_ACTIVITY NE **X**) AND (USED_OIL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**.) If **True**, set ERROR_FLAG = **ON**, Continue.	
--	USED_OIL_RCRA_DESC	--	--	This column will be deleted in the first translator guide revision after RCRAInfo is implemented. It is not being converted from RCRIS to RCRAInfo because there have been no values defined. It has been removed from the RCRAInfo HACTIVITY table.
--	USED_OIL_COLLECTION_SITE	–	--	No allowed value edit on this data column.
HB120	USED_OIL_COLLECTION_SITE, USED_OIL_RECYCLER_ACTIVITY	USED_OIL_RECYCLER_ACTIVITY must = X if USED_OIL_COLLECTION_SITE not blank.	(TRANS_CODE = **C** OR **Blank**) AND USED_OIL_COLLECTION_SITE NE **Blank** AND USED_OIL_RECYLER_ACTIVITY NE **X**. If **True**, set ERROR_FLAG = **ON**, Continue.	
--	USED_OIL_STATE_STATUS	–	–	No edit on this data column
--	USED_OIL_STATE_DESC	–	–	No edit on this data column.
HB130	HWFUEL_ACTIVITY	HWFUEL_ACTIVITY must = B, D, E, N, X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_ACTIVITY NE **B** OR **D** OR **E** OR **N** OR **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB132	HWFUEL_MARKET_BURNER	HWFUEL_MARKET_BURNER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_MARKET_BURNER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB133	HWFUEL_MARKET_BURNER, HWFUEL_ACTIVITY	HWFUEL_ACTIVITY must = B, D, E, N, or X if HWFUEL_MARKET_BURNER = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_MARKET_BURNER = **X** AND HWFUEL_ACTIVITY NE **B** OR **D** OR **E** OR **N** OR **X.** If **True**, set ERROR_FLAG = **ON**, Continue.	
HB134	HWFUEL_OTHER_MARKETER	HWFUEL_OTHER_MARKETER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_OTHER_MARKETER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB135	HWFUEL_OTHER_MARKETER, HWFUEL_ACTIVITY	HWFUEL_ACTIVITY must = B, D, E, N, or X if HWFUEL_OTHER_MARKETER = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_OTHER_MARKETER = **X** AND (HWFUEL_ACTIVITY NE **B** OR **D** OR **E** OR **N** OR **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB136	HWFUEL_BURNER	HWFUEL_BURNER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_BURNER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON,** Continue.	
HB137	HWFUEL_BURNER, HWFUEL_ACTIVITY	HWFUEL_ACTIVITY must = X if HWFUEL_BURNER = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_BURNER = **X** AND HWFUEL_ACTIVITY NE **X**. If **True**, set ERROR_FLAG = **ON**, Continue.	
HB140	HWFUEL_UTILITY_BOILER	HWFUEL_UTILITY_BOILER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_UTILITY_BOILER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB141	HWFUEL_UTILITY_BOILER, HWFUEL_ACTIVITY, HWFUEL_BURNER	HWFUEL_ACTIVITY must = X and HWFUEL_BURNER must = X if HWFUEL_UTILITY_BOILER = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_UTILITY_BOILER = **X** AND (HWFUEL_ACTIVITY NE **X** OR HWFUEL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB142	HWFUEL_INDUSTRIAL_BOILER	HWFUEL_INDUSTRIAL_BOILER must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_INDUSTRIAL_BOILER NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB143	HWFUEL_INDUSTRIAL_BOILER, HWFUEL_ACTIVITY, HWFUEL_BURNER	HWFUEL_ACTIVITY must = X and HWFUEL_BURNER must = X if HWFUEL_INDUSTRIAL_BOILER = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_INDUSTRIAL_BOILER = **X** AND (HWFUEL_ACTIVITY NE **X** OR HWFUEL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB144	HWFUEL_INDUSTRIAL_FURNACE	HWFUEL_INDUSTRIAL_FURNACE must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (HWFUEL_INDUSTRIAL_FURNACE NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB145	HWFUEL_INDUSTRIAL_FURNACE, HWFUEL_ACTIVITY, HWFUEL_BURNER	HWFUEL_ACTIVITY must = X and HWFUEL_BURNER must = X if HWFUEL_INDUSTRIAL_FURNACE = X.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_INDUSTRIAL_FURNACE = **X** AND (HWFUEL_ACTIVITY NE **X** OR HWFUEL_BURNER NE **X**). If **True**, set ERROR_FLAG = **ON,** Continue.	
HB150	HWFUEL_RCRA_STATUS, HWFUEL_ACTIVITY	HWFUEL_RCRA_STATUS must = R, P, A, or N.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_ACTIVITY = **X**) AND USED_OIL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB152	HWFUEL_RCRA_STATUS, HWFUEL_ACTIVITY	HWFUEL_RCRA_STATUS must = R, P, A, N or blank.	(TRANS_CODE = **C** OR **Blank**) AND HWFUEL_ACTIVITY NE **X**) AND USED_OIL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**.) If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
--	HWFUEL_RCRA_DESC	–	--	This column will be deleted from the flat file specifications in the first translator guide revision after RCRAInfo is implemented. It is not being converted from RCRIS to RCRAInfo because there have been no values defined. It has been removed from the RCRAInfo HACTIVITY table.
--	HWFUEL_STATE_STATUS	–	–	No edit on this data column
--	HWFUEL_STATE_DESC	–	–	No edit on this data column.
HB160	UNIVERSAL_ACTIVITY	UNIVERSAL_ACTIVITY must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (UNIVERSAL_ACTIVITY NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
HB162	UNIVERSAL_RCRA_STATUS, UNIVERSAL_ACTIVITY	UNIVERSAL_RCRA_STATUS must = R, P, A, or N.	(TRANS_CODE = **C** OR **Blank**) AND (UNIVERSAL_ACTIVITY = **X**) AND (UNIVERSAL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB164	UNIVERSAL_RCRA_STATUS, UNIVERSAL_ACTIVITY	TSD_RCRA_STATUS must = R, P, A, N, or blank.	(TRANS_CODE = **C** OR **Blank**) AND (USED_OIL_RECYCLER_ACTIVITY NE **X**) AND (USED_OIL_RCRA_STATUS NE **R** OR **P** OR **A** OR **N** OR **Blank**.) If **True**, set ERROR_FLAG = **ON**, Continue.	
--	UNIVERSAL_RCRA_DESC	--	--	This column will be deleted from the flat file specifications in the first translator guide revision after RCRAInfo is implemented. It is not being added because there have been no values defined. It has been removed from the RCRAInfo HACTIVITY table.
--	UNIVERSAL_STATE_STATUS	–	–	No edit on this data column.
--	UNIVERSAL_STATE_DESC	–	–	No edit on this data column.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HB170	UNDERGROUND_INJECTION_ACTIVITY	UNDERGROUND_INJECTION_ACTIVITY must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (UNDERGROUND_INJECTION_ACTIVITY NE **X** OR **Blank.**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HB180	RECYCLER_ACTIVITY	RECYCLER_ACTIVITY must = C, R, N or blank.	(TRANS_CODE = **C** OR **Blank**) AND (RECYCLER_ACTIVITY NE **C** OR **R** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON,** Continue.	
HB190	In HB flat file record: GENERATOR_ACTIVITY, TRANSPORTER_ACTIVITY TSD_ACTIVITY, USED_OIL_RECYCLER_ACTIVITY, HWFUEL_ACTIVITY, UNIVERSAL_ACTIVITY, UNDERGROUND_INJECTION_ACTIVITY, RECYCLER_ACTIVITY. In HHANDLER table: NON_NOTIFIER	At least one Activity Indicator must be marked.	(TRANS_CODE = **C** OR **Blank**) AND (NON_NOTIFIER for HANDLER_ID AND ACTIVITY_LOCATION in HHANDLER table NE **X** OR **E**) AND GENERATOR_ACTIVITY = **Blank** AND TRANSPORTER_ACTIVITY = **Blank** AND TSD_ACTIVITY = **Blank** AND USED_OIL_RECYCLER_ACTIVITY = **Blank** AND HWFUEL_ACTIVITY = **Blank** AND UNIVERSAL_ACTIVITY = **Blank** AND UNDERGROUND_INJECTION_ACTIVITY = **Blank** AND RECYCLER_ACTIVITY = **Blank**. If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	At least one Activity Indicator must be marked for each handler unless the Non-Notifier indicator in HHANDLER = X or E.
HHAZARDOUS_WASTE_STREAM (HC) Edits				
HC000	ACTIVITY_LOCATION (*FKHSOURCEFK_HHANDLERACTIVITY*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
HC010	SOURCE_TYPE	SOURCE_TYPE must = N, A, S, E, or R. Cannot determine IOR.	SOURCE_TYPE NE **N** OR **A** OR **S** OR **E** OR **R**. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HC012	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*)	Not IOR for HSOURCE block for ACTIVITY_LOCATION &activity_location and SOURCE_TYPE &source_type.	Match ACTIVITY_LOCATION in HC record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HSOURCE**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching SOURCE_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HSOURCE is part of a variable IOR block with HHANDLER. HSOURCE data can have one owner per ACTIVITY_LOCATION and SOURCE_TYPE.
HC014	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*)	Ref. Integrity Viol. (HSOURCE) Cannot process HHAZARDOUS_WASTE_STREAM record.	Row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE **NOT** found in HSOURCE table If **True,** reject record.	Referential integrity check.
HC016	WASTE_STREAM_SEQUENCE	WASTE_STREAM_SEQUENCE must be numeric	WASTE_STREAM_SEQUENCE not numeric. If **True**, reject record.	
HC018	HANDLER_ID (*FK_HSOURCEFK_HANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), WASTE_STREAM_SEQUENCE.	HHAZARDOUS_WASTE_STREAM row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE exists in HHAZARDOUS_WASTE_STREAM table. If **True**, reject record.	
HC020	HANDLER_ID (*FK_HSOURCEFK_HHANDLERFK_HBASIC*), ACTIVITY_LOCATION (*FK_HSOURCEFK_HHANDLERACTIVITY*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), WASTE_STREAM_S=UENCE	HHAZARDOUS_WASTE_STREAM row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE **NOT** found in HHAZARDOUS_WASTE_STREAM table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HC030	WASTE_UNIT_OF_MEASURE, WASTE_STREAM_AMOUNT	WASTE_UNIT_OF_MEASURE must = P, K, M, or T.	(TRANS_CODE = **C** OR **Blank**) AND WASTE_STREAM_AMOUNT NE **Blank** AND (WASTE_UNIT_OF_MEASURE NE **P** OR **K** OR **M** OR **T**). If **True**, set ERROR_FLAG = **ON**, Continue.	If a waste stream amount is provided, a valid unit of measure must also be provided.
HC032	WASTE_UNIT_OF_MEASURE, WASTE_STREAM_AMOUNT	WASTE_STREAM_AMOUNT cannot be blank.	(TRANS_CODE = **C** or **Blank**) AND WASTE_STREAM_AMOUNT = **Blank** AND (WASTE_UNIT_OF_MEASURE NE **Blank**) If **True**, set ERROR_FLAG = **ON**, Continue.	If a waste unit of measure is provided, a waste stream amount must be provided.
HC034	WASTE_UNIT_OF_MEASURE	WASTE_UNIT_OF_MEASURE must = P, K, M, T or blank.	(TRANS_CODE = **C** or **Blank**) AND (WASTE_UNIT_OF_MEASURE NE **P** OR **K** OR **M** OR **T** OR **Blank**) If **True**, set ERROR_FLAG = **ON**, Continue.	
HC036	WASTE_STREAM_AMOUNT	WASTE_STREAM_AMOUNT not numeric.	(TRANS_CODE = **C** or **Blank**) AND (WASTE_STREAM_AMOUNT not numeric). If **True**, set ERROR_FLAG = **ON**, Continue	
HC038	WASTE_STREAM_AMOUNT, WASTE_UNIT_OF_MEASURE	WASTE_STREAM_AMOUNT too large.	(TRANS_CODE = **C** or **Blank**) AND (WASTE_UNIT_OF_MEASURE = **P** OR **K** OR **T**) AND ((WASTE_STREAM_AMOUNT GT 999999999.) OR (WASTE_UNIT_OF_MEASURE = **T** AND WASTE_STREAM_AMOUNT GT 907194048.80000)). If **True,** set ERROR_FLAG = **ON**, Continue.	If Waste Unit of Measure is Pounds (P), Kilograms (K), or Tons (T), the whole portion of the number before the decimal point cannot be greater than nine (9) digits. If Waste Unit of Measure is Metric Tons (T), the associated amount cannot be greater than 907194048.80000).
–	WASTE_STREAM_DESC	–	–	There are no edits on this data column.
HC050	WASTE_STREAM_DATE	WASTE_STREAM_DATE invalid.	(TRANS_CODE = **C** or **Blank**) AND WASTE_STREAM_DATE not valid or **Blank**.) If True, set ERROR_FLAG = **ON**. If ERROR_FLAG = **ON,** reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HHAZARDOUS_WASTE_CODE (HD) Edits				
HD000	ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSO URCEFK_HH*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
HD010	SOURCE_TYPE	SOURCE_TYPE must = N, A, S, E, or R. Cannot determine IOR.	SOURCE_TYPE NE **N** OR **A** OR **S** OR **E** OR **R**. If **True**, reject record.	
HD012	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSO URCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSO URCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSO URCESOURC*)	Not IOR for HSOURCE block for ACTIVITY_LOCATION &activity_location and SOURCE_TYPE &source_type.	Match ACTIVITY_LOCATION in HD record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HSOURCE**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching SOURCE_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HSOURCE is part of a variable IOR block with HHANDLER. HSOURCE data can have one owner per ACTIVITY_LOCATION and SOURCE_TYPE.
HD014	WASTE_STREAM_SEQUENCE	WASTE_STREAM_SEQUEN CE must be numeric	WASTE_STREAM_SEQUENCE not numeric. If **True**, reject record.	
HD016	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSO URCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSO URCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSO URCESOURC*),WASTE_STREA M_SEQUENCE (*FK_HHAZARDOUS_WWASTE_ STREAM_SE*)	Ref. Integrity Viol (HHAZARDOUS_WASTE_S TREAM; Cannot process HHAZARDOUS_WASTE_C ODE row.	Row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE **NOT** found in HHAZARDOUS_WASTE_STREAM table If **True,** reject record.	Referential integrity check.
HD018	OWNER (*FK_LU_HAND_WASTOWNER)*	WASTE_CODE Owner must = US, or translator's Region or State code.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER NE **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	The identity of the translator is determined from information received via variable provided as part of load submittal process.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HD019	WASTE_CODE (*FK_LU_HAND_WASTWASTE_CODE*), OWNER (*FK_LU_HAND_WASTOWNER)*	WASTE_CODE not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND WASTE_CODE for OWNER **NOT** found in LU_HAND_WASTE table. If **True**, reject record.	
HD020	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSOURCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSOURCESOURC*), WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*), OWNER (*FK_LU_HAND_WASTOWNER*), WASTE_CODE (*FK_LU_HAND_WASTWASTE_CODE*).	HHAZARDOUS_WASTE_CODE row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE AND OWNER AND WASTE_CODE exists in HHAZARDOUS_WASTE_CODE table. If **True**, reject record.	
HD022	HANDLER_ID (*FK_HHANDLERFK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*), OWNER (*FK_LU_HAND_WASTOWNER*), WASTE_CODE (*FK_LU_HAND_WASTWASTE_CODE*).	HHAZARDOUS_WASTE_CODE row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE AND OWNER AND WASTE_CODE **NOT** found in HHAZARDOUS_WASTE_CODE table. If **True**, reject record.	
HHAZARDOUS_WASTE_PROCESS (HE) Edits				
HE000	ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH)*	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
HE010	SOURCE_TYPE	SOURCE_TYPE must = N, A, S, E, or R. Cannot determine IOR.	SOURCE_TYPE NE **N** OR **A** OR **S** OR **E** OR **R**. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HE012	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSOURCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSOURCESOURC*)	Not IOR for HSOURCE block for ACTIVITY_LOCATION &activity_location and SOURCE_TYPE &source_type.	Match ACTIVITY_LOCATION in HC record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HSOURCE**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching SOURCE_TYPE in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	HSOURCE is part of a variable IOR block with HHANDLER. HSOURCE data can have one owner per ACTIVITY_LOCATION and SOURCE_TYPE.
HE014	WASTE_STREAM_SEQUENCE	WASTE_STREAM_SEQUENCE must be numeric	WASTE_STREAM_SEQUENCE not numeric. If **True**, reject record.	
HE016	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSOURCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSOURCESOURC*),WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*)	Ref. Integrity Viol (HHAZARDOUS_WASTE_STREAM; Cannot process HHAZARDOUS_WASTE_PROCESS row.	Row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE **NOT** found in HHAZARDOUS_WASTE_STREAM table. If **True,** reject record.	Referential integrity check.
HE018	OWNER (*FK_LU_HWASTE_PROWNER*)	PROCESS_CODE Owner must = US, or translator's Region or State code.	(TRANS_CODE = **C** OR **Blank**) AND (OWNER NE **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	The identity of the translator is determined from information received via variable provided as part of load submittal process.
HE019	PROCESS_CODE (*FK_LU_HWASTE_PRPROCESS_CODE*), OWNER (*FK_LU_HWASTE_PROWNER*)	PROCESS_CODE not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND PROCESS_CODE for OWNER **NOT** found in LU_HWASTE_PROCESS_CODE table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
HE020	HANDLER_ID (*FK_HHAZARDOUS_WFK_HSOURCEFK_HH*), ACTIVITY_LOCATION (*FK0HHAZARDOUS_WFK_HSOURCEFK_HH*), SOURCE_TYPE (*FK_HHAZARDOUS_WFK_HSOURCESOURC*), WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*), OWNER (*FK_LU_HWASTE_PROWNER*), PROCESS_CODE (*FK_LU_HAND_HWASTE_PRPROCESS_CODE*).	HHAZARDOUS_WASTE_PROCESS row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE AND OWNER AND PROCESS_CODE exists in HHAZARDOUS_WASTE_PROCESS table. If **True**, reject record.	
HE022	HANDLER_ID (*FK_HHANDLERFK_HBASIC_HANDLER_ID*), ACTIVITY_LOCATION (*FKHHANDLERACTIVITY_LOCATION*), SOURCE_TYPE (*FK_HSOURCESOURCE_TYPE*), WASTE_STREAM_SEQUENCE (*FK_HHAZARDOUS_WWASTE_STREAM_SE*), OWNER (*FK_LU_HWASTE_PROWNER*), PROCESS_CODE (*FK_LU_HAND_HWASTE_PRPROCESS_CODE*).	HHAZARDOUS_WASTE_PROCESS_CODE row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND ACTIVITY_LOCATION AND SOURCE_TYPE AND WASTE_STREAM_SEQUENCE AND OWNER AND PROCESS_CODE **NOT** found in HHAZARDOUS_WASTE_PROCESS table. If **True**, reject record.	
PSERIES (P1) Edits				
P1010	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Not IOR for PSERIES block for &activity_location.	Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = PSERIES. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	PSERIES data can have one owner per state prefix. PSERIES data is fixed IOR.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P1012	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol (HBASIC); Cannot process PSERIES record.	HANDLER_ID NOT in HHANDLER table. (Use first two characters of HANDLER_ID to determine ACTIVITY_LOCATION). If **True,** reject record.	Referential integrity check.
P1014	SERIES_SEQ	SERIES_SEQ must be numeric	SERIES_SEQ not numeric. If **True**, reject record.	
P1016	HANDLER_ID (*FK_HBASICHANDLER_ID*), SERIES_SEQ	PSERIES row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND SERIES_SEQ exists in PSERIES. If **True**, reject record.	
P1018	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), SERIES_SEQ	PSERIES row missing; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND SERIES_SEQ **NOT** found in PSERIES. If **True**, reject record.	
P1020	SERIES_NAME	SERIES_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND SERIES_NAME = **Blank**.. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other fields in record. At END, If ERROR_FLAG = ON, reject record.
P1030	EPAOWNER (*FK_LU_STAFF_EPAOWNER*), EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*)	EPAPERSON_ID Owner must = translator's Region code.	(TRANS_CODE = **C** or **Blank**) AND EPAPERSON_ID NE **Blank** and EPAOWNER NE Region code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to STATEOWNER edit (P1040)	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
P1031	EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*), EPAOWNER (*FK_LU_STAFF_EPAOWNER*)	EPAPERSON_ID not in LU table.	(TRANS_CODE = **C** or **Blank**) AND EPAPERSON_ID NE **Blank** AND EPAPERSON_ID for EPAOWNER NOT found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue.	
P1040	STATEOWNER (*FK_LU_STAFF_STATEOWNER*)	STATEPERSON_ID Owner must = State code being processed	(TRANS_CODE = **C** or **Blank**) AND STATEPERSON_ID NE **Blank** AND OWNER NE State code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue to EPAATTYOWNER edit (P1050).	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P1041	STATEPERSON_ID (*FK_LU_STAFF_STATEPERSON_ID*)	STATEPERSON._ID not in LU table.	(TRANS_CODE = **C** or **Blank**) AND STATEPERSON ID NE **Blank** AND STATEPERSON_ID for STATEOWNER **NOT** found in LU_STAFF table. If True, set ERROR_FLAG = **ON,** Continue.	
P1050	EPAATTYOWNER (*FK_LU_STAFF_EPAATTYOWNER*), EPAATTYPERSON_ID (*FK_LU_STAFF_EPAATTYPERSON_ID*)	EPAATTYOWNER Owner must = translator's Region code.	(TRANS_CODE = **C** or **Blank**) AND EPAATTYPERSON_ID NE **Blank** AND EPAATTYOWNER NE Region code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to STATEATTYOWNER edit (P1060)	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
P1051	EPAATTYPERSON_ID (*FK_LU_STAFF_EPAATTYPERSON_ID*)	EPAATTYPERSON_ID not in LU table	(TRANS_CODE = **C** or **Blank**) AND EPAATTYPERSON_ID NE **Blank** AND EPAATTYPERSON_ID **NOT** found in LU_STAFF table.	
P1060	STATEATTYOWNER (*FK_LU_STAFF_STATEATTYOWNER*)	STATEATTYPERSON_ID Owner must = State code being processed	(TRANS_CODE = **C** or **Blank**) AND STATEATTYPERSON_ID NE **Blank** AND STATEATTYOWNEROWNER NE State code for state being processed. If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	
P1061	STATEATTYPERSON_ID (*FK_LU_STAFF_STATEATTYPERSON_ID*)	STATEATTYPERSON._ID not in LU table.	(TRANS_CODE = **C** or **Blank**) AND STATEATTYPERSON ID NE **Blank** AND STATEATTYPERSON_ID for STATEATTYOWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	
--	NOTES	–	--	No edit on this data column.
PEVENT (P2) Edits				
P2000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If True, reject record.	ACTIVITY_LOCATION must be same as state being processed.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P2010	RESPONSIBLE_AGENCY	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	RESPONSIBLE_AGENCY NE **E** OR **S** OR **J.** If **True**, reject record.	
P2012	HANDLER_ID (*FK_PSERIESFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY.	Not IOR for AEVENT block for ACTIVITY_LOCATION &activity_location and PMT Event RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in P2 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD where TABLENAME in TSIMPLEMENTER_OF_RECORD = **PEVENT**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	PEVENT is a variable IOR block. The decision column for PEVENT is RESPONSIBLE_AGENCY. PEVENT data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
P2014	HANDLER_ID (*FK_PSERIESFK_HBASIC_HANDLER_ID*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC). Cannot process PEVENT row.	Row matching HANDLER_ID NOT found in HBASIC or TR_HBASIC table. If **True**, reject record.	Referential integrity check.
P2016	SERIES_SEQ	SERIES_SEQ must be numeric	SERIES_SEQ not numeric. If **True**, reject record.	
P2018	HANDLER_ID (*FK_PSERIESFK_HBASIC_HANDLER_ID*)	Matching row not found in PSERIES; Cannot process PEVENT.	Row matching HANDLER_ID AND SERIES_SEQ **NOT** found in PSERIES table. If **True**, reject record.	
P2020	OWNER (*FK_LU_PERMIT_EVOWNER*)	Event OWNER must = HQ, US, or &region_code.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = **E** AND OWNER NE **HQ** OR **US** OR Region Code for translator submitting job. If **True**, reject record.	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
P2021	OWNER (*FK_LU_PERMIT_EVOWNER*)	Event OWNER must = HQ, US, or &state_code.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = **S** AND OWNER NE **HQ** OR **US** OR State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the P2 CONTROL File.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P2022	OWNER (*FK_LU_PERMIT_EVOWNER*)	Event OWNER must = HQ, US, or IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = J AND OWNER NE **HQ** OR **US** OR IOR owner of **J**. If **True**, reject record.	
P2023	PERMIT_EVENT_CODE (*FK_LU_PERMIT_EVPERMIT_EVENT_CO*)	PERMIT_EVENT_CODE not in LU table.	PERMIT_EVENT_CODE for Event OWNER not found in LU_PERMIT_EVENT table. If **True**, reject record.	
P2030	EVENT_SEQ	EVENT_SEQ not numeric	EVENT_SEQ NOT numeric. If **True**, reject record.	
P2032	HANDLER_ID (*FK_PSERIES_FK_HBASICHANDLER_ID*), SERIES_SEQ (*FK_PSERIESSERIES_SEQ*), OWNER (*FK_LU_PERMIT_EVOWNER*), PERMIT_EVENT_CODE (*FK_LU_PERMTI_EVPERMIT_EVENT_CO*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, EVENT_SEQ.	PEVENT row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND SERIES_SEQ AND Pmt Event OWNER AND PERMIT_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ exists in PEVENT table. If **True**, reject record.	
P2034	HANDLER_ID (*FK_PSERIES_FK_HBASICHANDLER_ID*), SERIES_SEQ (*FK_PSERIESSERIES_SEQ*), OWNER (*FK_LU_PERMIT_EVOWNER*), PERMIT_EVENT_CODE (*FK_LU_PERMTI_EVPERMIT_EVENT_CO*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, EVENT_SEQ.	PEVENT row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND SERIES_SEQ AND Pmt Event OWNER AND PERMIT_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ **NOT** found in PEVENT table. If **True**, reject record.	
P2040	ACTUAL_DATE	Invalid ACTUAL_DATE.	(TRANS_CODE EQ **C** OR **Blank)** AND SCHEDULE_DATE_ORIG EQ **Blank** AND SCHEDULE_DATE_NEW EQ **Blank** AND (ACTUAL_DATE NOT valid date OR ACTUAL_DATE GT current date). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P2042	SCHEDULE_DATE_ORIG	Invalid SCHEDULE_DATE_ORIG	(TRANS_CODE EQ **C** OR **Blank**) AND SCHEDULE_DATE_ORIG NE **Blank** AND SCHEDULE_DATE_ORIG **NOT** valid date GT 05/18/1980.) If **True**, set ERROR_FLAG = **ON**, Continue.	
P2044	SCHEDULE_DATE_NEW	Invalid SCHEDULE_DATE_NEW	(TRANS_CODE EQ **C** OR **Blank**) AND SCHEDULE_DATE_NEW NE **Blank** AND SCHEDULE_DATE_NEW **NOT** valid date GT 05/18/1980.) If **True**, set ERROR_FLAG = **ON**, Continue.	
P2050	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue .	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
P2051	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue.	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
P2052	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND Suborganization OWNER NE IOR owner of **J.** If **True**, set ERROR_FLAG = **ON**, Continue.	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J**.
P2053	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND PERSON_IDENTIFIER for OWNER not found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue. If ERROR_FLAG = **ON**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
--	NOTES	–	–	No edit on this data element.
PUNIT (P3)				
P3010	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Not IOR for PUNIT block for &activity_location.	Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = PUNIT. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	PUNIT data can have one owner per state prefix. PUNIT data is fixed IOR.
P3012	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol (HBASIC); Cannot process PUNIT record.	Row matching HANDLER_ID NOT in HBASIC table. If **True,** reject record.	Referential integrity check.
P3014	UNIT_SEQ	UNIT_SEQ must be numeric	UNIT_SEQ not numeric. If **True**, reject record.	
P3016	HANDLER_ID (*FK_HBASICHANDLER_ID*), UNIT_SEQ	PUNIT row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND UNIT_SEQ exists in PUNIT. If **True**, reject record.	
P3018	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), UNIT_SEQ	PUNIT row missing; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND UNIT_SEQ **NOT** found in PUNIT. If **True**, reject record.	
P3020	UNIT_NAME	UNIT_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND UNIT_NAME = **Blank**.. If **True**, reject record.	
–	NOTES	–	–	No edit on this data column.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
PUNIT_DETAIL (P4) Edits				
P4010	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Not IOR for PUNIT block for &activity_location.	Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = PUNIT. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	PUNIT data can have one owner per state prefix. PUNIT data is fixed IOR.
P4012	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol (HBASIC); Cannot process PUNIT_DETAIL record.	Row matching HANDLER_ID **NOT** found in HBASIC table. If **True,** reject record.	Referential integrity check.
P4014	UNIT_SEQ	UNIT_SEQ must be numeric	UNIT_SEQ not numeric. If **True**, reject record.	
P4016	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), UNIT_SEQ	Ref. Integrity Viol. (PUNIT); Cannot process PUNIT_DETAIL record.	Row matching HANDLER_ID AND UNIT_SEQ **NOT** found in PUNIT table. If **True**, reject record.	
P4020	UNIT_DETAIL_SEQ	UNIT_DETAIL_SEQ must be numeric.	UNIT_DETAIL_SEQ not numeric. If **True**, reject record.	
P4040	HANDLER_ID (*FK_PUNITFK_HBASICHANDLER_ID*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*), UNIT_DETAIL_SEQ	PUNIT_DETAIL row exists; Cannot add.	TRANS_CODE = **Blank** AND row matching HANDLER_ID AND UNIT_SEQ AND UNIT_DETAIL_SEQ exist in PUNIT_DETAIL table. If **True**, reject record.	
P4042	HANDLER_ID (*FK_PUNITFK_HBASICHANDLER_ID*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*), UNIT_DETAIL_SEQ	PUNIT_DETAIL row not found; Cannot delete.	TRANS_CODE = **D** AND row matching HANDLER_ID AND UNIT_SEQ AND UNIT_DETAIL_SEQ **NOT** found in PUNIT_DETAIL table. If **True**, reject record.	
P4050	OWNER (*FK_LU_PROCESS_COWNER*)	Process Code OWNER must = HQ.	(TRANS_CODE = **C** or **Blank**) AND OWNER NE **HQ** If **True**, set ERROR_FLAG = **ON**, Continue to EFFECTIVE_DATE edit (P4060)	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P4051	PROCESS_CODE (*FK_LU_PROCESS_CPROCESS_CODE*)	PROCESS_CODE not in LU table.	(TRANS_CODE = **C** or **Blank**) AND PROCESS_CODE for OWNER NOT found in LU_PROCESS_CODE table. If **True,** set ERROR FLAG = **ON,** Continue to EFFECTIVE_DATE edit (P4060)	
P4052	OWNER (*FK_LU_PROCESS_CFK_LU_UOMOWNER*)	Process UOM must = HQ	(TRANS_CODE = **C** or **Blank**) AND PROCESS UOM OWNER NE **HQ**. If **True,** set ERROR_FLAG = **ON,** Continue to EFFECTIVE_DATE edit (P4060)	
P4053	UNIT_OF_MEASURE_TYPE (*FK_LU_PROCESS_CFK_LU_UOM_TYPE*)	Invalid Process UOM Type	(TRANS_ CODE = **C** or **Blank**) AND UOM_TYPE for PROCESS_CODE NOT found in LU_UNIT_OF_MEASURE table. If **True,** set ERROR_FLAG = **ON**, Continue.	
P4060	EFFECTIVE_DATE	EFFECTIVE_DATE cannot be blank.	(TRANS_CODE = **C** or **Blank**) AND EFFECTIVE_DATE = **Blank.** If **True**, set ERROR_FLAG = **ON,** Continue to CAPACITY edit (P4070).	
P4062	EFFECTIVE_DATE	EFFECTIVE_DATE must be valid date not greater than current date.	(TRANS_CODE = **C** or **Blank**) AND (EFFECTIVE_DATE **NOT** valid date OR EFFECTIVE_DATE GT CURRENT_DATE). If **True**, set ERROR_FLAG = **ON**, Continue.	
P4070	CAPACITY	CAPACITY cannot be blank.	(TRANS_CODE = **C** or **Blank**) AND CAPACITY = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue to CAPACITY TYPE OWNER edit (P4080)	
P4072	CAPACITY	CAPACITY must be numeric	(TRANS_CODE = **C** or **Blank**) AND CAPACITY **NOT** numeric. If **True**, set ERROR_FLAG = ON, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P4080	OWNER (*FK_LU_CAPACITY_OWNER*)	CAPACITY_TYPE Owner must = **US**.	(TRANS_CODE = **C** or **Blank**) AND CAPACITY_TYPE NE **Blank** AND CAPACITY_TYPE OWNER NE **US**. If **True**, set ERROR_FLAG = **ON**, Continue to NUMBER_OF_UNITS edit (P4090)**.**	
P4081	CAPACITY_TYPE (*FK_LU_CAPACITY_CAPACITY_TYPE*)	CAPACITY_TYPE not in LU table.	(TRANS_CODE = **C** or **Blank**) AND CAPACITY_TYPE NE **Blank** AND CAPACITY_TYPE for OWNER **NOT** in LU_CAPACITY_TYPE table. If **True**, set ERROR_FLAG = **ON**, Continue.	
P4090	NUMBER_OF_UNITS	NUMBER_OF_UNITS cannot be blank.	(TRANS_CODE = **C** or **Blank**) AND NUMBER_OF_UNITS = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue to COMMERCIAL_STATUS_CODE OWNER edit (P4100).	
P4091	NUMBER_OF_UNITS	NUMBER_OF_UNITS not numeric.	(TRANS_CODE = **C** or **Blank**) AND NUMBER_OF_UNITS **NOT** numeric. If **True**, set ERROR_FLAG = **ON**, Continue.	
P4100	OWNER (*FK_LU_COMMERCIAOWNER*)	COMMERCIAL_STATUS OWNER must = HQ.	(TRANS_CODE = **C** or **Blank**) AND COMMERCIAL_STATUS_CODE NE **Blank** AND COMMERCIAL_STATUS OWNER NE **HQ**. If **True**, set ERROR_FLAG = **ON**, Continue to LEG_OP_STATUS_CODE OWNER edit (P4110).	
P4101	COMMERCIAL_STATUS_CODE (*FK_LU_COMMERCIACOMMERCIAL_STAT*)	COMMERCIAL_STATUS_CODE not in LU table.	(TRANS_CODE = **C** or **Blank**) AND COMMERCIAL_STATUS_CODE NE **Blank** AND COMMERICAL_STATUS_CODE for OWNER **NOT** in LU_COMMERCIAL_STATUS table. If **True,** set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P4110	OWNER (*FK_LU_LEGALOPEOWNER*)	LEG_OP_STATUS_CODE Owner must = HQ	(TRANS_CODE = **C** or **Blank**) AND LEG_OP_STATUS_CODE OWNER NE **HQ**. If **True**, set ERROR_FLAG = **ON**, Continue.	
P4111	LEG_OP_STATUS_CODE (*FK_LU_LEGAL_OPELEG_OP_STATUS_C*)	LEG_OP_STATUS_CODE not in LU table.	(TRANS_CODE = **C** or **Blank**) AND LEG_OP_STATUS_CODE for OWNER **NOT** in LU_LEGAL_OPERATING_STATUS table. If **True**, set ERROR_FLAG = **ON**, Continue.	
P4120	OWNER (*FK_LU_SUBPROCESSOWNER*)	SUBPROCESS_CODE OWNER must = US	(TRANS_CODE = **C** or **Blank**) AND SUBPROCESS_CODE NE **Blank** AND SUBPROCESS_CODE OWNER NE **US.** If **True,** set ERROR_FLAG = **ON**. If ERROR_FLAG = **ON**, reject record.	
P4121	SUBPROCESS_CODE (*FK_LU_SUBPROCESSSUBPROCESS_CODE*)	SUBPROCESS_CODE not in LU table.	(TRANS_CODE = **C** or **Blank**) AND SUBPROCESS_CODE NE **Blank** AND SUBPROCESS_CODE for OWNER NOT in LU_SUBPROCESS_CODE table. If **True**, set ERROR_FLAG = **ON.** If ERROR_FLAG = **ON**, reject record.	
--	NOTES	–	–	No edit on this data column.
PLN_EVENT_UNIT (P5) Edits				
P5000	ACTIVITY_LOCATION (*FK_PEVENTACTIVITY_LOCATION*)	Invalid ACTIVITY_LOCATION	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If True, reject record.	ACTIVITY_LOCATION must be same as state being processed.
P5010	RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_AGENCY*)	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	RESPONSIBLE_AGENCY NE **E** OR **S** OR **J.** If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P5012	HANDLER_ID (*FK_PEVENTFK_PSERIESFK_H BASICHA*), ACTIVITY_LOCATION (*FK_PEVENTACTIVITY_LOCATI ON*), RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_A GENCY*).	Not IOR for AEVENT block for ACTIVITY_LOCATION &activity_location and PMT Event RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in P5 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD where TABLENAME in TSIMPLEMENTER_OF_RECORD = **PEVENT**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	PEVENT is a variable IOR block. The decision column for PEVENT is RESPONSIBLE_AGENCY. PEVENT data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
P5016	SERIES_SEQ (*FK_PEVENTFK_PSERIESSERI ES_SEQ*)	SERIES_SEQ must be numeric	SERIES_SEQ not numeric. If **True**, reject record.	
P5018	HANDLER_ID (*FK_PEVENTFK_PSERIESFK_H GASICHA*), SERIES_SEQ (*FK_PEVENTFK_PSERIESSERI ES_SEQ*)	Matching row not found in PSERIES; Cannot process PLN_EVENT_UNIT link or unlink.	Row matching HANDLER_ID AND SERIES_SEQ **NOT** found in PSERIES table. If **True**, reject record.	Referential integrity violation.
P5020	OWNER (*FK_PEVENTFK_LU_PERMIT_E VOWNER*)	Event OWNER must = HQ, US, or &region_code.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = **E** AND OWNER NE **HQ** OR **US** OR Region Code for translator submitting job. If **True**, reject record.	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
P5021	OWNER (*FK_PEVENTFK_LU_PERMIT_E VOWNER*)	Event OWNER must = HQ, US, or &state_code.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = **S** AND OWNER NE **HQ** OR **US** OR State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the P2 CONTROL File.
P5022	OWNER (*FK_PEVENTFK_LU_PERMIT_E VOWNER*)	Event OWNER must = HQ, US, or IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND Pmt Event RESPONSIBLE_AGENCY = **J** AND OWNER NE **HQ** OR **US** OR IOR owner of **J**. If **True**, reject record.	
P5023	PERMIT_EVENT_CODE (*FK_PEVENTFK_LU_PERMIT_E VPERMIT*)	PERMIT_EVENT_CODE not in LU table.	PERMIT_EVENT_CODE for Event OWNER not found in LU_PERMIT_EVENT table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P5030	EVENT_SEQ (*FK_PEVENTEVENT_SEQ*)	EVENT_SEQ not numeric	EVENT_SEQ NOT numeric. If **True**, reject record.	
P5040	UNIT_SEQ (*FK_PUNITUNIT_SEQ*)	UNIT_SEQ not numeric	UNIT_SEQ NOT numeric. If **True**, reject record.	
P5050	UNIT_DETAIL_SEQ (*FK_PUNIT_DETAILUNIT_DETAIL_SEQ*)	UNIT_DETAIL_SEQ not numeric.	UNIT_DETAIL_SEQ not numeric. If **True**, reject record.	
P5060	HANDLER_ID (*FK_PEVENTFK_PSERIES_FK_HBASICHA*), SERIES_SEQ (*FKPEVENTFK_PSERIESSERIES_SEQ*), OWNER (*FK_PEVENTFK_LU_PERMIT_EVOWNER*), PERMIT_EVENT_CODE (*FK_PEVENTFK_LU_PERMTI_EVPERMIT*), ACTIVITY_LOCATION (*FK_PEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_PEVENTEVENT_SEQ*).	PEVENT row not found; Cannot process PLN_EVENT_UNIT link or unlink.	Row matching HANDLER_ID AND SERIES_SEQ AND Pmt Event OWNER AND PERMIT_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ **NOT** found in PEVENT table. If **True**, reject record.	
P5062	HANDLER_ID (*FK_PUNITFK_HBASICHANDLER_ID*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*)	PUNIT row not found; Cannot process PLN_EVENT_UNIT link or unlink.	Row matching HANDLER_ID AND UNIT_SEQ **NOT** found in PUNIT table. If **True**, reject record.	
P5064	HANDLER_ID (*FK_PUNITFK_HBASICHANDLER_ID*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*), UNIT_DETAIL_SEQ (*FK_PUNIT_DETAILUNIT_DETAIL_SEQ*)	PUNIT_DETAIL row not found; Cannot process PLN_EVENT_UNIT link or unlink.	Row matching HANDLER_ID AND UNIT_SEQ AND UNIT_DETAIL_SEQ **NOT** found in PUNIT_DETAIL table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P5070	HANDLER_ID (*FK_PEVENT_FK_PSERIESFK_HBASICHA* and *FK_PUNITFK_HBASICHANDLER_ID*), SERIES_SEQ (*FKPEVENTFK_PSERIESSERIES_SEQ*), OWNER (*FK_PEVENTFK_LU_PERMITEVOWNER*), PERMIT_EVENT_CODE (*FK_PEVENTFK_LU_PERMIT_EVPERMIT*), ACTIVITY_LOCATION (*FK_PEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_PEVENTEVENT_SEQ*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*), UNIT_DETAIL_SEQ (*FK_PUNIT_DETAILUNIT_DETAIL_SEQ)*	PLN_EVENT_UNIT row exists; Cannot create link.	TRANS_CODE = **Blank** AND HANDLER_ID AND SERIES_SEQ AND Pmt Event OWNER AND PERMIT_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ AND UNIT_SEQ AND UNIT_DETAIL_SEQ exist in PLN_EVENT_UNIT table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
P5072	HANDLER_ID (*FK_PEVENT_FK_PSERIESFK_HBASICHA* and *FK_PUNITFK_HBASICHANDLER_ID*), SERIES_SEQ (*FKPEVENTFK_PSERIESSERIES_SEQ*), OWNER (*FK_PEVENTFK_LU_PERMITEVOWNER*), PERMIT_EVENT_CODE (*FK_PEVENTFK_LU_PERMIT_EVPERMIT*), ACTIVITY_LOCATION (*FK_PEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_PEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_PEVENTEVENT_SEQ*), UNIT_SEQ (*FK_PUNITUNIT_SEQ*), UNIT_DETAIL_SEQ (*FK_PUNIT_DETAILUNIT_DETAIL_SEQ)*	PLN_EVENT_UNIT row not found; Cannot delete link.	TRANS_CODE = **D** AND HANDLER_ID AND SERIES_SEQ AND Pmt Event OWNER AND PERMIT_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ AND UNIT_SEQ AND UNIT_DETAIL_SEQ exist in PLN_EVENT_UNIT table. If **True**, reject record.	
AAREA (A1) Edits				
A1010	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Not IOR for AAREA block for &activity_location.	Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD = AAREA. Determine if AUTHORIZED_AGENCY NE owner group for Translator. If **True**, reject record.	AAREA data can have one owner per state prefix. AAREA data is fixed IOR.
A1012	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol (HBASIC); Cannot process AAREA record.	HANDLER_ID NOT in HBASIC. (Use first two characters of HANDLER_ID to determine ACTIVITY_LOCATION). If **True,** abort translator load.	Referential integrity check.
A1014	AREA_SEQ	AREA_SEQ must be numeric	AREA_SEQ not numeric. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A1016	HANDLER_ID (*FK_HBASICHANDLER_ID*), AREA_SEQ	AAREA row exists; Cannot add.	TRANS_CODE = **Blank** AND (HANDLER_ID AND AREA_SEQ) exist in AAREA. If **True**, reject record.	
A1018	HANDLER_ID (*FK_HBASIC_HANDLER_ID*), AREA_SEQ	AAREA row missing; Cannot delete.	TRANS_CODE = **D** AND HANDLER_ID AND AREA_SEQ **NOT** found in AAREA. If **True**, reject record.	
A1020	AREA_NAME	AREA_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND AREA_NAME = **Blank**. If **True**, set ERROR_FLAG = **ON**, Continue.	If field fails edit, software applies remaining edits to other fields in record. At END, If ERROR_FLAG = ON, reject record.
A1030	AREA_FACILITYWIDE_IND	AREA_FACILITYWIDE_IND Must = Y or N.	(TRANS_CODE = **C** OR **Blank**) AND (AREA_FACILITYWIDE_IND NE **Y** OR **N**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A1040	AREA_NAME	AREA_NAME cannot be blank.	(TRANS_CODE = **C** OR **Blank**) AND (AREA_NAME = **Blank**) If **True**, set ERROR_FLAG = **ON**, Continue.	
A1050	AIR_RELEASE_IND	AIR_RELEASE_IND Must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (AIR_RELEASE_IND NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A1052	GROUNDWATER_RELEASE_IND	GROUNDWATER_RELEASE_IND Must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (GROUNDWATER_RELEASE_IND NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A1054	SOIL_RELEASE_IND	SOIL_RELEASE_IND Must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (SOIL_RELEASE_IND NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A1056	SURFACE_WATER_RELEASE_IND	SURFACE_WATER_RELEASE_IND Must = X or blank.	(TRANS_CODE = **C** OR **Blank**) AND (SURFACE_WATER_RELEASE_IND NE **X** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A1060	REGULATED_UNIT_IND	REGULATED_UNIT_IND Must = Y or N or blank.	(TRANS_CODE = **C** OR **Blank**) AND (REGULATED_UNIT_IND NE **Y** OR **N** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A1070	EPAOWNER (*FK_LU_STAFF_EPAOWNER*), EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*)	EPAPERSON_ID Owner must = translator's Region code.	(TRANS_CODE = **C** OR **Blank**) AND EPAPERSON_ID NE **Blank** AND EPAOWNER NE Region code for translator submitting job. If **True**, set ERROR_FLAG = ON, Continue to STATEOWNER edit (A1080)	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A1071	EPAPERSON_ID (*FK_LU_STAFF_EPAPERSON_ID*), EPAOWNER (*FK_LU_STAFF_EPAOWNER*)	EPAPERSON_ID not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND EPAPERSON_ID NE **Blank** AND EPAPERSON_ID for EPAOWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue.	
A1080	STATEOWNER (*FK_LU_STAFF_EPAOWNER*), STATEPERSON_ID (*FK_LU_STAFF_STATEPERSON_ID*)	STATEPERSON_ID Owner must = State code being processed.	(TRANS_CODE = **C** OR **Blank**) AND STATEPERSON_ID NE **Blank** AND OWNER NE State code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the A1 CONTROL File.
A1081	STATEPERSON_ID (*FK_LU_STAFF_STATEPERSON_ID*), STATEOWNER (*FK_LU_STAFF_STATEOWNER*)	STATEPERSON_ID not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND STATEPERSON_ID NE **Blank** AND STATEPERSON_ID for STATEOWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**. If ERROR_FLAG = **ON**, reject record.	
--	NOTES	–	–	No edit on this data column.
ACA_AUTHORITY (A2) Edits				
A2000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
A2010	RESPONSIBLE_AGENCY	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True,** reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A2012	HANDLER_ID (*FK_HBASICHANDLER_ID),* ACTIVITY_LOCATION , RESPONSIBLE_AGENCY	Not IOR for ACA_AUTHORITY block for ACTIVITY_LOCATION &activity_location and Authority RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A2 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HCA_AUTHORITY**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	ACA_AUTHORITY is a variable IOR block. The decision column for ACA_AUTHORITY is RESPONSIBLE_AGENCY. ACA_AUTHORITY data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
A2014	HANDLER_ID (*FK_HBASICHANDLER_ID),*	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process ACA_AUTHORITY row.	HANDLER_ID **NOT** in HBASIC or TR_HBASIC If **True,** reject record..	Referential integrity check.
A2016	OWNER (*FK_LU_AUTHORITYOWNER*)	Authority OWNER must = HQ	Authority OWNER NE **HQ**. If **True**, reject record.	
A2017	AUTHORITY_TYPE (*FK_LU_AUTHORITYAUTHORIT Y_TYPE*)	AUTHORITY_TYPE not in LU table.	AUTHORITY_TYPE for Authority OWNER **NOT** found in LU_AUTHORITY table. If **True**, reject record..	
A2018	EFFECTIVE_DATE	EFFECTIVE_DATE invalid or blank.	EFFECTIVE_DATE not valid date. If **True**, reject record.	
A2020	HANDLER_ID (*FK_HBASICHANDLER_ID*), OWNER (*FK_LU_AUTHORITYOWNER*), AUTHORITY_TYPE (*FK_LU_AUTHORITYAUTHORIT Y_TYPE*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, EFFECTIVE_DATE.	ACA_AUTHORITY row exists; Cannot add.	TRANS_CODE = **Blank** AND HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE exist in ACA_AUTHORITY table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A2022	HANDLER_ID (*FK_HBASICHANDLER_ID*), OWNER (*FK_LU_AUTHORITYOWNER*), AUTHORITY_TYPE (*FK_LU_AUTHORITYAUTHORITY_TYPE*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY, EFFECTIVE_DATE.	ACA_AUTHORITY row not found; Cannot delete.	TRANS_CODE = **D** AND HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE **NOT** found in ACA_AUTHORITY table. If **True**, reject record.	
A2030	ISSUE_DATE	Invalid ISSUE_DATE	(TRANS_CODE = **C** OR **Blank**) AND (ISSUE_DATE **NOT** valid date or **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A2040	REVOKE_DATE	Invalid REVOKE_DATE.	(TRANS_CODE = **C** OR **Blank**) AND (REVOKE_DATE **NOT** valid date or **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A2050	REPOSITORY	REPOSITORY must = 1, 2, 3 or blank.	(TRANS_CODE = **C** OR **Blank**) AND (REPOSITORY NE **1** OR **2** OR **3** OR **Blank**). If **True**, set ERROR_FLAG = **ON**, Continue.	
A2060	OWNER (*FK_LU_LEAD_PROGOWNER*)	Lead Program OWNER must = HQ.	(TRANS_CODE = **C** OR **Blank**) AND LEAD_PROGRAM NE **Blank** AND OWNER NE **HQ**). If True, set ERROR_FLAG = **ON**, Continue to Suborganization OWNER edit (A2070).	
A2061	LEAD_PROGRAM (*FK_LU_LEAD_PROGLEAD_PROGRAM*)	LEAD_PROGRAM. not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND LEAD_PROGRAM NE **Blank** AND LEAD_PROGRAM for OWNER not found in LU_LEAD_PROGRAM table. If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A2070	OWNER (*FK_LU_SUBORGANISUBORGANIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A2071	OWNER (*FK_LU_SUBORGANISUBORGANIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A2072	OWNER (*FK_LU_SUBORGANISUBORGANIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = IOR owner of 'J' Resp. Agnecy.	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND Suborganization OWNER NE IOR owner of **J.** If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J.**
A2073	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION*)	SUBORGANIZATION not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND SUBORGANIZATION for OWNER not found in LU_SUBORGANIZATION table. If **True**, set ERROR_FLAG = **ON**, Continue.	
A2080	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A2081	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A2082	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND Suborganization OWNER NE IOR owner of **J.** If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J.**
A2083	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENTIFIER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND PERSON_IDENTIFIER for OWNER not found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue.	
A2090	OWNER (*FK_LU_STAFF_ATTYOWNER*)	Attorney OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND ATTYPERSON_ID NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, reject record..	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A2091	OWNER (*FK_LU_STAFF_ATTYOWNER*)	Attorney OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND ATTYPERSON_ID NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A2092	OWNER (*FK_LU_STAFF_ATTYOWNER*)	Attorney OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND ATTYPERSON_ID NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND OWNER NE IOR owner of **J.** If **True**, reject record.	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J.**

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A2093	ATTYPERSON_ID (*FK_LU_STAFF_ATTYPERSON_ID*)	ATTYPERSON_ID not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND ATTYPERSON_ID NE **Blank** AND ATTYPERSON_ID for OWNER not found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue. If ERROR_FLAG = **ON**, reject record.	
–	NOTES	–	–	No edit on this data column.
ASTATUTE_CITATION (A3) Edits				
A3000	ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_LOCATI*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
A3010	RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSIBLE_AGE*)	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True,** reject record.	
A3012	HANDLER_ID (*FK_ACA_AUTHORFKFK_HBASI CHANDLE),* ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_LOCATI*) , RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSIBLE_AGE*)	Not IOR for ACA_AUTHORITY block for ACTIVITY_LOCATION &activity_location and Authority RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A3 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HCA_AUTHORITY**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	ASTATUTE_CITATION is part of a variable IOR block with ACA_AUTHORITY. The decision column for ACA_AUTHORITY is RESPONSIBLE_AGENCY. ASTATUTE_CITATION data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
A3014	OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*)	Authority OWNER must = HQ	Authority OWNER NE **HQ**. If **True**, reject record.	
A3015	AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*)	AUTHORITY_TYPE not in LU table.	AUTHORITY_TYPE for Authority OWNER **NOT** found in LU_AUTHORITY table. If **True**, reject record..	
A3016	EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIVE_DATE*)	EFFECTIVE_DATE invalid or blank.	EFFECTIVE_DATE not valid date. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A3018	HANDLER_ID (*FK_ACA_AUTHORITFK_HBASI CHANDLE),* OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_ LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIV E_DATE*)	Ref. Integrity Viol. (ACA_AUTHORITY); Cannot process ASTATUTE_CITATION row.	HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE **NOT** in ACA_AUTHORITY table. If **True,** reject record..	Referential integrity check.
A3020	OWNER (*FK_LU_STATUTORYOWNER*)	Citation OWNER must = US.	Statutory Citation OWNER NE **US**. If **True**, reject record.	
A3021	STATUTORY_CITATION (*FK_LU_STATUTORYSTATUTO RY_CITAT*)	STATUTORY_CITATION Not in LU table.	STATUTORY_CITATION for Statutory Citation OWNER NOT in LU_STATUTORY table. If **True**, reject record	
A3030	HANDLER_ID (*FK_ACA_AUTHORITFK_HBASI CHANDLE),* OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_ LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIV E_DATE*), OWNER (*FK_LU_STATUTORYOWNER*), STATUTORY_CITATION (*FK_LU_STATUTORYSTATUTO RY_CITAT*)	ASTATUTE_CITATION row exists; Cannot add.	TRANS_CODE = **Blank** AND HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE AND Statutory Citation OWNER AND STATUTORY_CITATION exist in ASTATUTE_CITATION table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A3032	HANDLER_ID (*FK_ACA_AUTHORITFK_HBASI CHANDLE*), OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_ LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIV E_DATE*), OWNER (*FK_LU_STATUTORYOWNER*), STATUTORY_CITATION (*FK_LU_STATUTORYSTATUTO RY_CITAT*)	ASTATUTE_CITATION row not found; Cannot delete.	TRANS_CODE = **D** AND HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE AND Statutory Citation OWNER AND STATUTORY_CITATION **NOT** found in ASTATUTUE_CITATION table. If **True**, reject record.	
–	NOTES	–	–	No edit on this data column.
AEVENT (A4) Edits				
A4000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
A4010	RESPONSIBLE_AGENCY (for Event)	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	CA Event RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True**, reject record.	
A4012	HANDLER_ID (*FK_HBASICHANDLER_ID),* ACTIVITY_LOCATION , RESPONSIBLE_AGENCY	Not IOR for AEVENT block for ACTIVITY_LOCATION &activity_location and CA Event RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A4 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **AEVENT**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	AEVENT is a variable IOR block. The decision column for AEVENT is RESPONSIBLE_AGENCY. AEVENT data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A4014	HANDLER_ID (*FK_HBASICHANDLER_ID*),	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process AEVENT row.	HANDLER_ID **NOT** in HBASIC or TR_HBASIC If **True,** reject record..	Referential integrity check.
A4016	OWNER (*FK_LU_CA_EVENTOWNER*)	Event OWNER must = HQ, US or &region_code.	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **E** AND OWNER NE **HQ** OR **US** OR Region Code for translator submitting job. If **True**, reject record..	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A4017	OWNER (*FK_LU_CA_EVENTOWNER*)	Event OWNER must = HQ, US or &state_code	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **S** AND OWNER NE **HQ** OR **US** OR State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A4018	OWNER (*FK_LU_CA_EVENTOWNER*)	Event OWNER must = HQ OR US OR IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **J** AND OWNER NE HQ OR US OR IOR owner of **J.** If **True**, reject record.	Where CA Event RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J.**
A4019	CA_EVENT_CODE (*FK_LU_CA_EVENTCA_EVENT_ CODE*)	CA_EVENT_CODE not in LU table.	CA_EVENT_CODE for Event OWNER not found in LU_CA_EVENT table. If **True**, reject record.	
A4020	EVENT_SEQ	EVENT_SEQ not numeric.	EVENT_SEQ NOT numeric. If **True**, reject record.	
A4022	HANDLER_ID (*FK_HBASICHANDLER_ID*), OWNER (*FK_LU_CA_EVENTOWNER*), CA_EVENT_CODE (*FK_LU_CAEVENTCA_EVENT_ CODE*), ACTIVITY_LOCATION, RESPONSIBLE_AGENCY (for Event), EVENT_SEQ.	AEVENT row exists; Cannot add.	TRANS_CODE = **Blank** AND HANDLER_ID AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ exist in AEVENT table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A4024	HANDLER_ID (*FK_HBASICHANDLER_ID*), OWNER (*FK_LU_CA_EVENTOWNER*), CA_EVENT_CODE (*FK_LU_CAEVENTCA_EVENT_ CODE)*, ACTIVITY_LOCATION, RESPONSIBLE_AGENCY (for Event), EVENT_SEQ.	AEVENT row not found; Cannot delete.	TRANS_CODE = **D** AND HANDLER_ID AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ **NOT** found in AEVENT table. If **True**, reject record.	
A4030	RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*)	RESPONSIBLE_AGENCY (for Authority) must = E, S, or J.	RESPONSIBLE_AGENCY (for Authority) NE **E** OR **S** OR **J**. If **True,** reject record.	An event row must be linked to an authority row.

A state can link an event to an authority for which the region is IOR; a region can link an event to an authority for which the state is IOR. |
A4040	ACTUAL_DATE, SCHEDULE_DATE_ORIG, SCHEDULE_DATE_NEW	Scheduled or Actual Event date must be provided.	(TRANS_CODE = **C** OR **Blank**) AND (SCHEDULE_DATE_ORIG EQ Blank AND SCHEDULE DATE_ NEW EQ Blank AND ACTUAL_DATE EQ Blank). If **True**, set ERROR_FLAG = **ON**, Continue.	
A4042	ACTUAL_DATE	Invalid ACTUAL_DATE	(TRANS_CODE = **C** OR **Blank**) AND ACTUAL_DATE NE Blank AND (ACTUAL_DATE NE valid date OR ACTUAL_DATE GREATER THAN the CURRENT_DATE) If **True**, set ERROR_FLAG = **ON**, Continue.	
A4044	SCHEDULE_DATE_ORIG	Invalid SCHEDULE_DATE_ORIG	(TRANS_CODE = **C** OR **Blank**) AND SCHEDULED_DATE_ORIG NE Blank AND SCHEDULED_DATE_ORIG NE valid date If **True**, set ERROR_FLAG = **ON**, Continue.	
A4046	SCHEDULE_DATE_NEW	Invalid SCHEDULE_DATE_NEW	(TRANS_CODE = **C** OR **Blank**) AND SCHEDULED_DATE_NEW NE Blank AND SCHEDULED_DATE_NEW NE valid date If **True**, set ERROR_FLAG = **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A4050	OWNER (*FK_LU_SUBORGANISUBORGA NIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A4051	OWNER (*FK_LU_SUBORGANISUBORGA NIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A4052	OWNER (*FK_LU_SUBORGANISUBORGA NIZATION*), RESPONSIBLE_AGENCY	SUBORGANIZATION OWNER must = IOR owner of 'J' Resp. Agnecy.	(TRANS_CODE = **C** OR **Blank**) AND SUBORGANIZATION NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND Suborganization OWNER NE IOR owner of **J.** If **True**, set ERROR_FLAG = **ON**, Continue to Staff OWNER edit (A2080).	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J.**
A4060	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A4061	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A4062	OWNER (*FK_LU_STAFFOWNER*), RESPONSIBLE_AGENCY	Staff OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND RESPONSIBLE_AGENCY = **J** AND Suborganization OWNER NE IOR owner of **J.** If **True**, set ERROR_FLAG = **ON**, Continue to Attorney OWNER edit (A2090).	Where RESPONSIBLE_AGENCY = **J,** code owner is based on the IOR owner of **J**.
A4064	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON_IDENT IFIER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND PERSON_IDENTIFIER for OWNER not found in LU_STAFF table. If **True**, set ERROR_FLAG = **ON**, Continue.	
–	NOTES	–	–	No edit on this data column.
ALN_AREA_EVENT (A5) Edits				
A5000	ACTIVITY_LOCATION (*FK_AEVENT_ACTIVITY_LOCAT ION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
A5010	RESPONSIBLE_AGENCY (*FK_AEVENT_RESPONSIBLE_A GENCY*)	RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	CA Event RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True,** reject record.	
A5012	HANDLER_ID (*FK_AEVENTfK_HBASICHANDL ER_ID),* ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATI ON*) , RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_A GENCY*)	Not IOR for AEVENT block for ACTIVITY_LOCATION &activity_location and CA Event RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A4 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **AEVENT**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	AEVENT is a variable IOR block. The decision column for AEVENT is RESPONSIBLE_AGENCY. AEVENT data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A5014	OWNER (*FK_AVENTFK_LU_CA_EVENT OWNER*)	Event OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, reject record..	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A5015	OWNER (*FK_AEVENTFK_LU_CA_EVENT OWNER*)	Event OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A5016	OWNER (*FK_AEVENTFK_LU_CA_EVENT OWNER*)	Event OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND CA Event RESPONSIBLE_AGENCY = **J** AND OWNER NE IOR owner of **J**. If **True**, reject record.	Where CA Event RESPONSIBLE_AGENCY = **J**, code owner is based on the IOR owner of **J**.
A5018	CA_EVENT_CODE (*FK_AVENTFK_LU_CA_EVENT CA_EVEN*)	CA_EVENT_CODE not in LU table.	CA_EVENT_CODE for Event OWNER not found in LU_CA_EVENT table. If **True**, reject record.	
A5020	EVENT_SEQ (*FK_AEVENTEVENT_SEQ*)	EVENT_SEQ not numeric.	EVENT_SEQ NOT numeric. If **True**, reject record.	
A5030	AREA_SEQ (*FK_AAREAAREA_SEQ*)	AREA_SEQ must be numeric	AREA_SEQ not numeric. If **True**, reject record.	
A5040	HANDLER_ID (*FK_AEVENTFK_HBASICHANDLER_ID*), OWNER (*FK_AVENTFK_LU_CAEVENTOWNER*), CA_EVENT_CODE (*FK_AVENTFK_LU_CAEVENTCA_EVEN*), ACTIVITY_LOCATION (*FK_AVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_AEVENT_EVENT_SEQ*)	AEVENT row missing. Cannot process ALN_AREA_EVENT link or unlink.	HANDLER_ID AND OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ **NOT** found in AEVENT. If **True**, reject record.	
A5042	HANDLER_ID (*FK_AAREAFK_HBASICHANDLER_ID*), AREA_SEQ	AAREA row missing; Cannot process ALN_AREA_EVENT link or unlink.	HANDLER_ID AND AREA_SEQ **NOT** found in AAREA. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A5050	HANDLER_ID (*FK_AAREAFK_HBASICHANDLER_ID* and *FKA_AEVENTFK_HBASICHANDLER_ID*), AREA_SEQ (*FK_AAREAAREA_SEQ*),OWNER (*FK_LU_CA_EVENTOWNER*), CA_EVENT_CODE (*FK_LU_CAEVENTCA_EVENT_CODE*), ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_AVENTEVENT_SEQ*).	ALN_AREA_EVENT row exists; Cannot create link.	TRANS_CODE = **Blank** AND HANDLER_ID AND AREA_SEQ AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ exist in ALN_AREA_EVENT table. If **True**, reject record.	
A5052	HANDLER_ID (*FK_AAREAFK_HBASICHANDLER_ID* and *FKA_AEVENTFK_HBASICHANDLER_ID*), AREA_SEQ (*FK_AAREAAREA_SEQ*),OWNER (*FK_LU_CA_EVENTOWNER*), CA_EVENT_CODE (*FK_LU_CAEVENTCA_EVENT_CODE*), ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_AVENTEVENT_SEQ*).	ALN_AREA_EVENT row not found; Cannot unlink.	TRANS_CODE = **D** AND HANDLER_ID AND AREA_SEQ AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ **NOT** found in ALN_AREA_EVENT table. If **True**, reject record.	
ALN_EVENT_AUTHORITY (A6) Edits				
A6000	ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION* and *FK_ACA_AUTHORITACTIVIT_LOCATI)*	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NOT = 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
A6010	RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*)	Event RESPONSIBLE_AGENCY must = E, S, or J. Cannot determine IOR.	CA Event RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True,** reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A6012	HANDLER_ID (*FK_AEVENTFK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_AEVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*)	Not IOR for AEVENT block for ACTIVITY_LOCATION &activity_location and CA Event RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A4 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **AEVENT**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	AEVENT is a variable IOR block. The decision column for AEVENT is RESPONSIBLE_AGENCY. AEVENT data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
A6014	OWNER (*FK_AEVENTFK_LU_CAEVENT OWNER*)	Event OWNER must = &region_code.	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **E** AND OWNER NE Region Code for translator submitting job. If **True**, reject record..	The identity of the translator is determined from information received via variable provided as part of load submittal process. If the translator is a state, the region is determined from the state code.
A6015	OWNER (*FK_AEVENTFK_LU_CAEVENT OWNER*)	Event OWNER must = &state_code	(TRANS_CODE = **C** OR **Blank**) AND CA Event RESPONSIBLE_AGENCY = **S** AND OWNER NE State Code for state being processed. If **True**, reject record.	State code is determined from the first two characters of the FLATFILE_NAME in the A2 CONTROL File.
A6016	OWNER (*FK_AEVENTFK_LU_CAEVENT OWNER*)	Event OWNER must = IOR owner of 'J' Resp. Agency.	(TRANS_CODE = **C** OR **Blank**) AND PERSON_IDENTIFIER NE **Blank** AND CA Event RESPONSIBLE_AGENCY = **J** AND OWNER NE IOR owner of **J.** If **True**, reject record.	Where CA Event RESPONSIBLE_AGENCY = **J**, code owner is based on the IOR owner of **J**.
A6018	CA_EVENT_CODE (*FK_AEVENTFK_LU_CA_EVENT CA_EVEN*)	CA_EVENT_CODE not in LU table.	CA_EVENT_CODE for Event OWNER not found in LU_CA_EVENT table. If **True**, reject record.	
A6020	EVENT_SEQ (*FK_AEVENTEVENT_SEQ*)	EVENT_SEQ not numeric.	EVENT_SEQ **NOT** numeric. If **True**, reject record.	
A6030	RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSIBLE_AGE*)	CA Authority RESPONSIBLE_AGENCY must = E, S, or J.	RESPONSIBLE_AGENCY NE **E** OR **S** OR **J**. If **True,** reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A6032	HANDLER_ID (*FK_HBASICHANDLER_ID),* ACTIVITY_LOCATION , RESPONSIBLE_AGENCY	Not IOR for ACA_AUTHORITY block for ACTIVITY_LOCATION &activity_location and Authority RESPONSIBLE_AGENCY &responsible_agency.	Match ACTIVITY_LOCATION in A2 record against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where TABLENAME in TSIMPLEMENTER_OF_RECORD = **HCA_AUTHORITY**. Determine if AUTHORIZED_AGENCY NE owner group for Translator AND if RESPONSIBLE_CODE matching RESPONSIBLE_AGENCY in TSRESPONSIBLE_PARTY NOT owned by owner group for Translator. If **True**, reject record.	ACA_AUTHORITY is a variable IOR block. The decision column for ACA_AUTHORITY is RESPONSIBLE_AGENCY. ACA_AUTHORITY data can have one owner per ACTIVITY_LOCATION and RESPONSIBLE_AGENCY.
A6034	HANDLER_ID (*FK_HBASICHANDLER_ID),*	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process ACA_AUTHORITY row.	HANDLER_ID **NOT** in HBASIC or TR_HBASIC If **True,** reject record..	Referential integrity check.
A6036	OWNER (*FK_LU_AUTHORITYOWNER*)	Authority OWNER must = HQ	Authority OWNER NE **HQ**. If **True**, reject record.	
A6037	AUTHORITY_TYPE (*FK_LU_AUTHORITYAUTHORITY_TYPE*)	AUTHORITY_TYPE not in LU table.	AUTHORITY_TYPE for Authority OWNER **NOT** found in LU_AUTHORITY table. If **True**, reject record..	
A6038	EFFECTIVE_DATE	EFFECTIVE_DATE invalid or blank.	EFFECTIVE_DATE not valid date. If **True**, reject record.	
A6040	HANDLER_ID (*FK_AEVENTFK_HBASICHANDLER_ID*), OWNER (*FK_AVENTFK_LU_CAEVENTOWNER*), CA_EVENT_CODE (*FK_AVENTFK_LU_CAEVENTCA_EVEN*), ACTIVITY_LOCATION (*FK_AVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_AEVENT_EVENT_SEQ*)	AEVENT row missing. Cannot process ALN_EVENT_AUTHORITY link or unlink.	HANDLER_ID AND OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EVENT_SEQ **NOT** found in AEVENT. If **True**, reject record.	Referential Integrity check

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A6042	HANDLER_ID (*FK_ACA_AUTHORITFK_HBASI CHANDLE),* OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_ LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIV E_DATE*)	ACA_AUTHORITY row missing. Cannot process ALN_EVENT_AUTHORITY link or unlink.	HANDLER_ID AND Authority OWNER AND AUTHORITY_TYPE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY AND EFFECTIVE_DATE **NOT** in ACA_AUTHORITY table. If **True,** reject record..	Referential integrity check.
A6050	HANDLER_ID (*FK_AEVENTFK_HBASICHANDL ER_ID* and *FK_ACA_AUTHORITFK_HBASIC HANDLE*), OWNER (*FK_AVENTFK_LU_CAEVENTO WNER*), CA_EVENT_CODE (*FK_AVENTFK_LU_CAEVENTC A_EVEN*), ACTIVITY_LOCATION (*FK_AVENTACTIVITY_LOCATIO N*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_A GENCY*), EVENT_SEQ (*FK_AEVENT_EVENT_SEQ*) OWNER (*FK_ACA_AUTHORITFK_LU_AU THOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AU THTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_ LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSI BLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIV E_DATE*)	ALN_EVENT_AUTHORITY row exists; Cannot create link.	TRANS_CODE = **Blank** AND HANDLER_ID AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ and CA Authority OWNER and AUTHORITY_TYPE and CA Authority ACTIVITY_LOCATION and CA Authority RESPONSIBLE_AGENCY and EFFECTIVE_DATE exist in ALN_EVENT_AUTHORITY table. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
A5052	HANDLER_ID (*FK_AEVENTFK_HBASICHANDLER_ID* and *FK_ACA_AUTHORITFK_HBASICHANDLE*), OWNER (*FK_AVENTFK_LU_CAEVENTOWNER*), CA_EVENT_CODE (*FK_AVENTFK_LU_CAEVENTCA_EVEN*), ACTIVITY_LOCATION (*FK_AVENTACTIVITY_LOCATION*), RESPONSIBLE_AGENCY (*FK_AEVENTRESPONSIBLE_AGENCY*), EVENT_SEQ (*FK_AEVENT_EVENT_SEQ*) OWNER (*FK_ACA_AUTHORITFK_LU_AUTHOWNER*), AUTHORITY_TYPE (*FK_ACA_AUTHORITFK_LU_AUTHTYPE*), ACTIVITY_LOCATION (*FK_ACA_AUTHORITACTIVITY_LOCATI*), RESPONSIBLE_AGENCY (*FK_ACA_AUTHORITRESPONSIBLE_AGE*), EFFECTIVE_DATE (*FK_ACA_AUTHORITEFFECTIVE_DATE*)	ALN_EVENT_AUTHORITY row not found; Cannot unlink.	TRANS_CODE = **D** AND HANDLER_ID AND AREA_SEQ AND CA Event OWNER AND CA_EVENT_CODE AND ACTIVITY_LOCATION AND RESPONSIBLE_AGENCY (for CA Event), AND EVENT_SEQ **NOT** found in ALN_AREA_EVENT table. If **True**, reject record.	
CEVALUATION (C1) Edits				
C1000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record..	ACTIVITY_LOCATION must be same as state being processed.
C1002	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol. (HBASIC or TR_HBASIC); Cannot process CEVALUATION record.	HANDLER_ID NOT found in HBASIC.or TR_HBASIC If **True,** reject record.	Referential integrity check.
C1004	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, EVALUATION_DATE, EVALUATION_SEQ, AND AGENCY	CEVALUATION row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND row matching HANDLER_ID, ACTIVITY_LOCATION, EVALUATION_DATE, EVALUATION_SEQ AND AGENCY exists in CEVALUATION table. If **True**, reject record.	This checks for duplicate flat file records.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C1006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
C1008	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION EVALUATION_DATE, EVALUATION_SEQ AND AGENCY	CEVALUATION row not found; Cannot delete.	TRANS_CODE EQ **D** AND row matching HANDLER_ID, ACTIVITY_LOCATION, EVALUATION_DATE, EVALUATION_SEQ AND AGENCY **NOT** found in CEVALUATION table. If **True**, reject record.	This checks to ensure there is a row to delete.
C1010	EVALUATION_DATE	Must enter a valid Evaluation Date	Date not a valid date. If **True**, reject record.	
C1012	AGENCY	Must enter a valid Agency for the evaluation.	AGENCY not equal E, C, S, B or X. If **True**, reject record.	
C1014	EVALUATION_SEQ	Must enter an Evaluation sequence number.	Sequence number is blank. If **True**, reject record.	
C1016	FOUND_VIOLATION	Must enter a valid response for FOUND_VIOLATION	FOUND_VIOLATION not equal Y, N or U. If **True**, reject record.	
C1018	FOUND_VIOLATION	FOUND_VIOLATION is "Y", yet no violations are linked to this evaluation.	If "Y", must have a row in C3 and C4 flat file. If **True**, reject record.	
C1020	FOUND_VIOLATION	FOUND_VIOLATION is "N" or "U" yet violations ARE linked to this evaluation.	If "N" or "U", must NOT have a row in C3 and C4 flat file. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C1022	OWNER (*FK_LU_EVALUATIOREASOWNER*)	REASON_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C1024	REASON_TYPE (*FK_LU_EVALUATIOREASON_TYPE*), OWNER (*FK_LU_EVALUATIOREASOWNER*)	REASON_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND REASON_TYPE for OWNER **NOT** found in LU_EVALUATION_REASON table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C1026	OWNER (*FK_LU_EVALUATIOWNER*)	EVALUATION_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C1028	EVALUATION_TYPE (*FK_LU_EVALUATIOEVALUATION_TYPE*), OWNER (*FK_LU_EVALUATIOWNER*)	EVALUATION_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND EVALUATION_TYPE for OWNER **NOT** found in LU_EVALUATION_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C1030	OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C1032	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION*), OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND SUBORGANIZATION for OWNER **NOT** found in LU_SUBORGANIZATION table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C1034	OWNER (*FK_LU_STAFFOWNER*)	PERSON owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C1036	PERSON_IDENTIFIER (*FK_LU_STAFFPERSON _IDENTIFIER*), OWNER (*FK_LU_STAFFOWNER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PERSON_IDENTIFIER for OWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
CEVALUATION_AREA (C2) Edits				
C2000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1ˢᵗ two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C2002	HANDLER_ID (*FK_CEVALUATIONFK_HBASIC HANDLER*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CEVALUATION_AREA record.	Row matching HANDLER_ID NOT found in HBASIC or TR_HBASIC table. If **True,** reject record.	Referential integrity check.
C2004	HANDLER_ID (*FK_CEVALUATIONFK_HBASIC HANDLER*), ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_L OCATIO), EVALUATION_DATE (FK_CEVALUATIONEVALUATIO N_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATIO N_SEQ), AGENCY (FK_CEVALUATIONAGENCY) AND AREA_TYPE (FK_LU_EVALUATIOAREA_TYP E)	CEVALUATION_AREA row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND row matching HANDLER_ID (*FK_CEVALUATIONFK_HBASICHANDLER*), ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_LOCATIO), EVALUATION_DATE (FK_CEVALUATIONEVALUATION_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATION_SEQ), AGENCY (FK_CEVALUATIONAGENCY) AND AREA_TYPE (FK_LU_EVALUATIOAREA_TYPE) exists in CEVALUATION_AREA. If **True**, reject record.	This checks for duplicate flat file records.
C2006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C2008	HANDLER_ID (*FK_CEVALUATIONFK_HBASIC HANDLER*), ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_L OCATIO), EVALUATION_DATE (FK_CEVALUATIONEVALUATIO N_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATIO N_SEQ), AGENCY (FK_CEVALUATIONAGENCY) AND AREA_TYPE (FK_LU_EVALUATIOAREA_TYP E)	CEVALUATION_AREA row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CEVALUATIONFK_HBASICHANDLER*), ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_LOCATIO), EVALUATION_DATE (FK_CEVALUATIONEVALUATION_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATION_SEQ), AGENCY (FK_CEVALUATIONAGENCY) AND AREA_TYPE (FK_LU_EVALUATIOAREA_TYPE)) **NOT** found in CEVALUATION_AREA. If **True**, reject record.	This checks to ensure there is a row to delete.
C2010	EVALUATED	Evaluated must have a value of E, NE, NA or blank.	EVALUATED not equal E, NE, NA or blank. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C2012	OWNER (*FK_LU_EVALUATIOOWNER*)	AREA_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C2014	AREA_TYPE (*FK_LU_EVALUATIOAREA_TYP E*), OWNER (*FK_LU_EVALUATIOOWNER*)	AREA_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND AREA_TYPE for OWNER **NOT** found in LU_EVALUATION_VIOLATION_AREA table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
CVIOLATION (C3) Edits				
C3000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C3002	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CVIOLATION record.	HANDLER_ID NOT found in HBASIC.or TR_HBASIC If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C3004	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, DETERMINED_BY_AGENCY AND VIOLATION_SEQ	CVIOLATION row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, DETERMINED_BY_AGENCY, AND VIOLATION_SEQ) exist in CVIOLATION. If **True**, reject record.	This checks for duplicate flat file records.
C3006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
C3008	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, DETERMINED_BY_AGENCY AND VIOLATION_SEQ	CVIOLATION row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, DETERMINED_BY_AGENCY AND VIOLATION_SEQ) **NOT** found in CVIOLATION. If **True**, reject record.	This checks to ensure there is a row to delete.
C3010	DETERMINED_BY_AGENCY	DETERMINED_BY_AGENCY must be the same as EVALUATION AGENCY.	DETERMINED_BY_AGENCY not equal EVALUATION AGENCY. If **True**, reject record.	
C3012	VIOLATION_SEQ	VIOLATION_SEQ can not be blank.	VIOLATION_SEQ equal blank. If **True**, reject record.	
C3014	DETERMINED_DATE	DETERMINED_DATE must be entered unless the class is "P".	DETERMINED_DATE equal blank and CLASS_TYPE not "P". If **True**, reject record.	
C3016	DETERMINED_DATE	DETERMINED_DATE must be a valid date.	DETERMINED_DATE not a valid date. If **True**, reject record.	
C3018	DETERMINED_DATE	DETERMINED_DATE must be on or after the EVALUATION DATE.	DETERMINED_DATE less than EVALUATION DATE. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C3020	RESPONSIBLE_AGENCY	RESPONSIBLE_AGENCY must be the same as DETERMINED_BY_AGENCY.	RESPONSIBLE_AGENCY not equal DETERMINED_BY_AGENCY. If **True**, reject record.	
C3022	ACTUAL_RTC_DATE	ACTUAL_RTC_DATE must be a valid date.	ACTUAL_RTC_DATE not a valid date. If **True**, reject record.	
C3024	ACTUAL_RTC_DATE	ACTUAL_RTC_DATE must be on or after the DETERMINED_DATE.	ACTUAL_RTC_DATE less than DETERMINED_DATE. If **True**, reject record.	
C3026	CITATION	CITATION must be entered since CITATION_TYPE was entered.	If CITATION is blank and CITATION_TYPE is not blank. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C3028	OWNER (*FK_LU_EVALUATIOOWNER*)	AREA_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3030	AREA_TYPE (*FK_LU_EVALUATIOAREA_TYPE*), OWNER (*FK_LU_EVALUATIOOWNER*)	AREA_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND AREA_TYPE for OWNER **NOT** found in LU_EVALUATION_VIOLATION_AREA table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C3032	OWNER (*FK_LU_CLASSOWNER*)	CLASS_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3034	CLASS_TYPE (*FK_LU_CLASSCLASS_TYPE*), OWNER (*FK_LU_CLASSOWNER*)	CLASS_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND CLASS_TYPE for OWNER **NOT** found in LU_CLASS table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C3038	OWNER (*FK_LU_PRIORITYOWNER*)	PRIORITY_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3040	PRIORITY_TYPE (*FK_LU_PRIORITYPRIORITY_TYPE*), OWNER (*FK_LU_PRIORITYOWNER*)	PRIORITY_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PRIORITY_TYPE for OWNER **NOT** found in LU_PRIORITY table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C3042	OWNER (*FK_LU_RTC_QUALIOWNER*)	RTC_QUALIFIER_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3044	RTC_QUALIFIER_TYPE ((*FK_LU_RTC_QUALIRTC_QUALIFIER_T*), OWNER (FK_LU_RTC_QUALIOWNER)	RTC_QUALIFIER_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND RTC_QUALIFIER_TYPE for OWNER **NOT** found in LU_RTC_QUALIFIER table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C3046	RTC_QUALIFIER_TYPE ((*FK_LU_RTC_QUALIRTC_QUALIFIER_T*)	RTC_QUALIFIER_TYPE can not be blank if ACTUAL_RTC_DATE was entered.	RTC_QUALIFIER_TYPE is blank and ACTUAL_RTC_DATE is not blank. If **True**, reject record.	
C3048	OWNER (*FK_LU_CITATION_OWNER*)	CITATION owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3050	CITATION_TYPE ((*FK_LU_CITATION_CITATION_TYPE*), OWNER (FK_LU_CITATION_OWNER)	CITATION_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND CITATION_TYPE for OWNER **NOT** found in LU_CITATION_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C3052	CITATION_TYPE ((*FK_LU_CITATION_CITATION_TYPE*)	CITATION_TYPE must be entered since CITATION was entered.	If CITATION_TYPE is blank and CITATION is not blank. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C3054	OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3056	SUBORGANIZATION (*FK_LU_SUBORGANIZATIONAN ISUBORGANIZATION _TYPE*), OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND SUBORGANIZATION for OWNER **NOT** found in LU_SUBORGANIZATION table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C3058	OWNER (*FK_LU_STAFFOWNEROWNER*)	PERSON owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C3060	PERSON_IDENTIFIER (*FK_LU_STAFFOWNERPERSO N _IDENTIFIER*), OWNER (*FK_LU_STAFFOWNER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PERSON_IDENTIFIER for OWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
CLN_EVALUATION_VIOLATION (C4) Edits				
C4000	ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_L OCATIO AND FK_CVIOLATIONACTIVITY_LOC ATI*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C4002	HANDLER_ID (FK_CEVALUATIONFK_HBASIC HANDLER AND FK_CVIOLATIONFK_HBASICHA NDLER)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CVIOLATION record.	HANDLER_ID NOT found in HBASIC.or TR_HBASIC If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C4004	HANDLER_ID (FK_CEVALUATIONFK_HBASIC HANDLER AND FK_CVIOLATIONFK_HBASICHA NDLER), ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_L OCATIO AND FK_CVIOLATIONACTIVITY_LOC ATI*), EVALUATION_DATE (FK_CEVALUATIONEVALUATIO N_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATIO N_SEQ), AGENCY (FK_CEVALUATIONAGENCY), DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED _BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_S EQ)	CEVALAUATION_VIOLATIO N row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (FK_CEVALUATIONFK_HBASICHANDLER AND FK_CVIOLATIONFK_HBASICHANDLER), ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_LOCATIO AND FK_CVIOLATIONACTIVITY_LOCATI*), EVALUATION_DATE (FK_CEVALUATIONEVALUATION_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATION_SEQ), AGENCY (FK_CEVALUATIONAGENCY), DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)) exist in CEVALAUATION_VIOLATION. If **True**, reject record.	This checks for duplicate flat file records.
C4006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_L OCATIO AND FKCVIOLATIONACTIVITY_LOCA TIO*)	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C4008	HANDLER_ID (FK_CEVALUATIONFK_HBASIC HANDLER AND FK_CVIOLATIONFK_HBASICHA NDLER), ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_L OCATIO AND FK_CVIOLATIONACTIVITY_LOC ATI*), EVALUATION_DATE (FK_CEVALUATIONEVALUATIO N_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATIO N_SEQ), AGENCY (FK_CEVALUATIONAGENCY), DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED _BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_S EQ)	CEVALAUATION_VIOLATIO N row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (FK_CEVALUATIONFK_HBASICHANDLER AND FK_CVIOLATIONFK_HBASICHANDLER), ACTIVITY_LOCATION (*FK_CEVALUATIONACTIVITY_LOCATIO AND FK_CVIOLATIONACTIVITY_LOCATI*), EVALUATION_DATE (FK_CEVALUATIONEVALUATION_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATION_SEQ), AGENCY (FK_CEVALUATIONAGENCY), DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)) **NOT** found in CEVALAUATION_VIOLATION. If **True**, reject record.	This checks to ensure there is a row to delete.
C4012	HANDLER_ID (FK_CEVALUATIONFK_HBASIC HANDLER), ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_L OCATIO), EVALUATION_DATE (FK_CEVALUATIONEVALUATIO N_DATE), EVALUATION_SEQ (FK_CEVALUATIONEVALUATIO N_SEQ) AND AGENCY (FK_CEVALUATIONAGENCY)	Ref. Integrity Viol. (CEVALUATION and TR_CEVALUATION); Cannot process CLN_EVALUATION_VIOLAT ION record.	HANDLER_ID (FK_CEVALUATIONFK_HBASICHANDLER) and ACTIVITY_LOCATION (FK_CEVALUATIONACTIVITY_LOCATIO) and EVALUATION_DATE (FK_CEVALUATIONEVALUATION_DATE) and EVALUATION_SEQ (FK_CEVALUATIONEVALUATION_SEQ) and AGENCY (FK_CEVALUATIONAGENCY) NOT found in CEVALUATION or TR_CEVALUATION. If **True,** reject record.	Referential integrity check.
C4014	HANDLER_ID (FK_CVIOLATIONFK_HBASICHA NDLER), ACTIVITY_LOCATION (FKCVIOLATIONACTIVITY_LOC ATION, DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED _BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_S EQ)	Ref. Integrity Viol. (CVIOLATION and TR_CVIOLATION); Cannot process CLN_EVALUATION_VIOLAT ION record.	HANDLER_ID (FK_CEVALUATIONFK_HBASICHANDLER) and ACTIVITY_LOCATION (FK_CEVIOLATIONACTIVITY_LOCATIO) and DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) and VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ) NOT found in CVIOLATION or TR_CVIOLATION. If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CENFORCEMENT (C5) Edits				
C5000	ACTIVITY_LOCATION	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record..	ACTIVITY_LOCATION must be same as state being processed.
C5002	HANDLER_ID (*FK_HBASICHANDLER_ID*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CENFORCEMENT record.	HANDLER_ID NOT found in HBASIC or TR_HBASIC. If **True,** reject record.	Referential integrity check.
C5004	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, ENFORCEMENT_DATE, ENFORCEMENT_SEQ, AND AGENCY	CENFORCEMENT row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, ENFORCEMENT_DATE, ENFORCEMENT_SEQ, AND AGENCY) exist in CENFORCEMENT. If **True**, reject record.	This checks for duplicate flat file records.
C5006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
C5008	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, ENFORCEMENT_DATE, ENFORCEMENT_SEQ, AND AGENCY	CENFORCEMENT row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION, ENFORCEMENT_DATE, ENFORCEMENT_SEQ, AND AGENCY) **NOT** found in CENFORCEMENT. If **True**, reject record.	This checks to ensure there is a row to delete.
C5010	ENFORCEMENT_SEQ	ENFORCEMENT_SEQ must be entered.	ENFORCEMENT_SEQ equal blank. If **True**, reject record.	
C5012	ENFORCEMENT_DATE	ENFORCEMENT_DATE must be a valid date.	ENFORCEMENT_DATE not a valid date. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C5014	ENFORCEMENT_DATE	ENFORCEMENT_DATE must be less than or equal to today.	ENFORCEMENT_DATE greater then today. If **True**, reject record.	
C5016	AGENCY	ENFORCEMENT AGENCY must be E, S or X.	AGENCY not equal E, S, or X. If **True**, reject record.	
–	DOCKET	–	–	There are no edits on this data column.
–	ATTORNEY	–	–	There are no edits on this data column.
–	NOTES	–	–	There are no edits on this data column.
C5018	OWNER (FK_LU_ENFORCEMEOWNER)	ENFORCEMENT_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C5020	ENFORCEMENT_TYPE (*FK_LU_ENFORCEMENFORCEMENT_TYP*), OWNER (FK_LU_ENFORCEMEOWNER)	ENFORCEMENT_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND ENFORCEMENT_TYPE for OWNER **NOT** found in LU_ENFORCEMENT_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C5022	OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C5024	SUBORGANIZATION (*FK_LU_SUBORGANISUBORGANIZATION_TYPE*), OWNER (*FK_LU_SUBORGANIOWNER*)	SUBORGANIZATION not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND SUBORGANIZATION for OWNER **NOT** found in LU_SUBORGANIZATION table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C5026	OWNER (*FK_LU_STAFFOWNEROWNER*)	PERSON owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C5028	PERSON_IDENTIFIER (*FK_LU_STAFFOWNERPERSON _IDENTIFIER*), OWNER (*FK_LU_STAFFOWNER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PERSON_IDENTIFIER for OWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
CMEDIA (C6) Edits				
C6000	ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C6002	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CMEDIA record.	HANDLER_ID NOT found in HBASIC or TR_HBASIC If **True,** reject record.	Referential integrity check.
C6004	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORC EMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORC EMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND MEDIA_TYPE (FK_LU_MEDIA_TYPMEDIA_TY PE)	CMEDIA row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDL E*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI) , ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_D AT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_S EQ), AGENCY (FK_CENFORCEMENTAGENCY) AND MEDIA_TYPE (FK_LU_MEDIA_TYPMEDIA_TYPE)) exist in CMEDIA. If **True**, reject record.	This checks for duplicate flat file records.
C6006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C6008	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND MEDIA_TYPE (FK_LU_MEDIA_TYPMEDIA_TYPE)	CMEDIA row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND MEDIA_TYPE (FK_LU_MEDIA_TYPMEDIA_TYPE)) **NOT** found in CMEDIA. If **True**, reject record.	This checks to ensure there is a row to delete.
–	NOTES	–	–	There are no edits on this data column.
C6010	OWNER (FK_LU_MEDIA_TYPOWNER)	MEDIA_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C6012	MEDIA_TYPE (*FK_LU_MEDIA_TYPMEDIA_TYPE*), OWNER (FK_LU_MEDIA_TYPOWNER)	MEDIA_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND MEDIA_TYPE for OWNER **NOT** found in LU_MEDIA_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
CSEP (C7) Edits				
C7000	ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C7002	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CSEP record.	HANDLER_ID NOT found in HBASIC or TR_HBASIC If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C7004	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORC EMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORC EMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND SEP_SEQ	CSEP row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDL E*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI) , ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_D AT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_S EQ), AGENCY (FK_CENFORCEMENTAGENCY) AND SEP_SEQ) exist in CSEP. If **True**, reject record.	This checks for duplicate flat file records.
C7006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
C7008	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORC EMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORC EMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND SEP_SEQ	CSEP row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDL E*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI) , ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_D AT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_S EQ), AGENCY (FK_CENFORCEMENTAGENCY) AND SEP_SEQ) **NOT** found in CSEP. If **True**, reject record.	This checks to ensure there is a row to delete.
C7010	SEP_SEQ	SEP_SEQ must be entered.	SEP_SEQ equal blank. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
–	EXPENDITURE_AMOUNT	–	–	There are no edits on this data column.
C7011	SCHED_DATE, ACTUAL_DATE	Either SCHED_DATE or ACTUAL_DATE must be a valid date	SCHED_DATE not a valid date and ACTUAL_DATE not a valid date. If **True**, reject record.	
C7012	SCHED_DATE	SEP_SCHED_DATE must be a valid date.	SCHED_DATE not a valid date. If **True**, reject record.	
C7014	SCHED_DATE	SEP_SCHED_DATE must be on or after the ENFORCEMENT_DATE.	SCHED_DATE less than ENFORCEMENT_DATE. If **True**, reject record.	
C7016	ACTUAL_DATE	ACTUAL_DATE must be a valid date.	ACTUAL_DATE not a valid date. If **True**, reject record.	
C7018	ACTUAL_DATE	ACTUAL_DATE must be on or after the ENFORCEMENT_DATE.	ACTUAL_DATE less than ENFORCEMENT_DATE. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C7020	OWNER (FK_LU_SEP_TYPEOWNER)	SEP_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C7022	SEP_TYPE (*FK_LU_SEP_TYPESEPTYPE*), OWNER (FK_LU_SEP_TYPEOWNER)	SEP_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND SEP_TYPE for OWNER **NOT** found in LU_SEP_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C7024	OWNER (*FK_LU_STAFFOWNEROWNER*)	PERSON owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C7026	PERSON_IDENTIFIER (*FK_LU_STAFFOWNERPERSON _IDENTIFIER*), OWNER (*FK_LU_STAFFOWNER*)	PERSON_IDENTIFIER not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PERSON_IDENTIFIER for OWNER **NOT** found in LU_STAFF table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CPENALTY (C8) Edits				
C8000	ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C8002	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE*)	Ref. Integrity Viol. (HBASIC or TR_HBASIC); Cannot process CPENALTY record.	HANDLER_ID NOT found in HBASIC or TR_HBASIC If **True,** reject record.	Referential integrity check.
C8004	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY _LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORC EMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORC EMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND PENALTY_TYPE	CPENALTY row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDL E*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI) , ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_D AT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_S EQ), AGENCY (FK_CENFORCEMENTAGENCY) AND PENALTY_TYPE) exist in CPENALTY. If **True**, reject record.	This checks for duplicate flat file records.
C8006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C8008	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND PENALTY_TYPE	CPENALTY row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE*), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) AND PENALTY_TYPE) **NOT** found in CPENALTY. If **True**, reject record.	This checks to ensure there is a row to delete.
C8010	AMOUNT	PENALTY AMOUNT must be entered.	AMOUNT equal blank. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C8012	OWNER (FK_LU_PENALTY_TOWNER)	PENALTY_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C8014	PENALTY_TYPE (*FK_LU_PENALTY_TPENALTY_TYPE*), OWNER (FK_LU_PENALTY_TOWNER)	PENALTY_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PENALTY_TYPE for OWNER **NOT** found in LU_PEANLTY_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	
C8016	PENALTY_TYPE	ENFORCEMENT_TYPE 100-199 and 800-899 can not impose any PENALTY.	ENFORCEMENT_TYPE is 100-199 OR 800-899 AND PENALTY_TYPE is not blank. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C8018	PENALTY_TYPE	ENFORCEMENT_TYPE can not impose this type of PENALTY.	ENFORCEMENT_TYPE is 200-299 AND PENALTY_TYPE is not PA or FA. If **True**, reject record. ENFORCEMENT_TYPE is 300-399 AND PENALTY_TYPE is not FA, FC or CR. If **True**, reject record. ENFORCEMENT_TYPE is 400-499 AND PENALTY_TYPE is not PA. If **True**, reject record. ENFORCEMENT_TYPE is 500-599 AND PENALTY_TYPE is not FA. If **True**, reject record. ENFORCEMENT_TYPE is 600-699 AND PENALTY_TYPE is not FA, FC or CR. If **True**, reject record. ENFORCEMENT_TYPE is 700-799 AND PENALTY_TYPE is not FA. If **True**, reject record.	
CPAYMENT (C9) Edits				
C9000	ACTIVITY_LOCATION (FK_CPENALTYFK_CENFORCE MENTACTI)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
C9002	HANDLER_ID (*FK_CPENALTYFK_CENFORCE MENTFK_H*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CPAYMENT record.	HANDLER_ID NOT found in HBASIC.or TR_HBASIC If **True,** reject record.	Referential integrity check.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C9004	HANDLER_ID (*FK_CPENALTYFK_CENFORCEMENTFK_H*), ACTIVITY_LOCATION (FK_CPENALTYFK_CENFORCEMENTACTI), ENFORCEMENT_DATE (FK_CPENALTYFK_CENFORCEMENTDATE), ENFORCEMENT_SEQ (FK_CPEANLTYFK_CENFORCEMENTSEQ), AGENCY (FKCPENALTYFK_CENFORCEMENTAGEN) PENALTY_TYPE (FK_CPENALTYFK_LU_PENALTY_TPENA) AND PAYMENT_SEQ	CPAYMENT row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_CPENALTYFK_CENFORCEMENTFK_H*), ACTIVITY_LOCATION (FK_CPENALTYFK_CENFORCEMENTACTI), ENFORCEMENT_DATE (FK_CPENALTYFK_CENFORCEMENTDATE), ENFORCEMENT_SEQ (FK_CPEANLTYFK_CENFORCEMENTSEQ), AGENCY (FKCPENALTYFK_CENFORCEMENTAGEN) PENALTY_TYPE (FK_CPENALTYFK_LU_PENALTY_TPENA) AND PAYMENT_SEQ) exist in CPAYMENT. If **True**, reject record.	This checks for duplicate flat file records.
C9006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
C9008	HANDLER_ID (*FK_CPENALTYFK_CENFORCEMENTFK_H*), ACTIVITY_LOCATION (FK_CPENALTYFK_CENFORCEMENTACTI), ENFORCEMENT_DATE (FK_CPENALTYFK_CENFORCEMENTDATE), ENFORCEMENT_SEQ (FK_CPEANLTYFK_CENFORCEMENTSEQ), AGENCY (FKCPENALTYFK_CENFORCEMENTAGEN) PENALTY_TYPE (FK_CPENALTYFK_LU_PENALTY_TPENA) AND PAYMENT_SEQ	CPAYMENT row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CPENALTYFK_CENFORCEMENTFK_H*), ACTIVITY_LOCATION (FK_CPENALTYFK_CENFORCEMENTACTI), ENFORCEMENT_DATE (FK_CPENALTYFK_CENFORCEMENTDATE), ENFORCEMENT_SEQ (FK_CPEANLTYFK_CENFORCEMENTSEQ), AGENCY (FKCPENALTYFK_CENFORCEMENTAGEN) PENALTY_TYPE (FK_CPENALTYFK_LU_PENALTY_TPENA) AND PAYMENT_SEQ) AND PENALTY_TYPE) **NOT** found in CPAYMENT. If **True**, reject record.	This checks to ensure there is a row to delete.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
C9010	PAYMENT_SEQ	PAYMENT_SEQ must be entered.	PAYMENT_SEQ equal blank. If **True**, reject record.	
C9012	SCHED_DATE	PAYMENT SCHED_DATE must be a valid date.	SCHED_DATE not a valid date. If **True**, reject record.	
C9014	SCHED_DATE	PAYMENT SCHED_DATE must be on or after the ENFORCEMENT_DATE.	SCHED_DATE less than ENFORCEMENT_DATE. If **True**, reject record.	
C9016	SCHED_AMOUNT	SCHED_AMOUNT must be entered.	SCHED_AMOUNT equal blank. If **True**, reject record.	
C9018	PAID_DATE	PAID_DATE must be a valid date if PAID_AMOUNT is entered.	PAID_DATE not a valid date and PAID_AMOUNT is not blank. If **True**, reject record.	
C9020	PAID_DATE	PAID_DATE must be on or after the ENFORCEMENT_DATE.	PAID_DATE less than ENFORCEMENT_DATE. If **True**, reject record.	
C9022	PAID_AMOUNT	PAID_AMOUNT must be entered if PAID_DATE is entered.	PAID_AMOUNT equal blank and PAID_DATE is not blank. If **True**, reject record.	
–	NOTES	–	–	There are no edits on this data column.
C9024	OWNER (FK_CPENALTYFK_LU_PENALTY_TOWNE)	PENALTY_TYPE owner must = HQ, US, or translator's Region or State code.	(TRANS_CODE EQ **C** OR **Blank**) AND (OWNER NOT = **HQ** OR **US** OR Region or State code matching that of translator submitting job). If **True**, reject record.	Information received via variable provided as part of load submittal process.
C9026	PENALTY_TYPE (FK_CPENALTYFK_LU_PENALTY_TPENA) (*FK_LU_PENALTY_TPENALTY_TYPE*), OWNER (FK_CPENALTYFK_LU_PENALTY_TOWNE)	PENALTY_TYPE not in LU table.	(TRANS_CODE EQ **C** OR **Blank**) AND PENALTY_TYPE for OWNER **NOT** found in LU_PEANLTY_TYPE table. If **True**, set ERROR_FLAG EQ **ON**, Continue.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CLN_COMPLIANCE_SCHEDULE (CA) Edits				
CA000	ACTIVITY_LOCATION (*FK_CENFORCEMENT_LOCATI AND FK_CVIOLATIONACTIVITY_LOC ATION*)	Invalid ACTIVITY_LOCATION.	ACTIVITY_LOCATION NE 1st two characters of FLATFILE_NAME. If **True**, reject record.	ACTIVITY_LOCATION must be same as state being processed.
CA002	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE AND FK_CVIOLATIONFK_HBASICHA NDLER*)	Ref. Integrity Viol. (HBASIC and TR_HBASIC); Cannot process CLN_COMPLIANCE_SCHE DULE record.	HANDLER_ID NOT found in HBASIC or TR_HBASIC. If **True,** reject record.	Referential integrity check.
CA004	HANDLER_ID (*FK_CENFORCEMENTFK_HBAS ICHANDLE AND FK_CVIOLATIONFK_HBASICHA NDLER*), ACTIVITY_LOCATION (*FK_CENFORCEMENT_LOCATI AND FK_CVIOLATIONACTIVITY_LOC ATION*), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORC EMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORC EMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED _BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_S EQ)	CLN_COMPLIANCE_SCHE DULE row exists; Cannot add.	TRANS_CODE EQ **Blank**. AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDL E AND FK_CVIOLATIONFK_HBASICHANDLER*), ACTIVITY_LOCATION (*FK_CENFORCEMENT_LOCATI AND FK_CVIOLATIONACTIVITY_LOCATION*), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_D AT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_S EQ), AGENCY (FK_CENFORCEMENTAGENCY) DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)) exist in CLN_COMPLIANCE_SCHEDULE. If **True**, reject record.	This checks for duplicate flat file records.

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CA006	HANDLER_ID (*FK_HBASICHANDLER_ID*), ACTIVITY_LOCATION *(FKCENFORCEMENT_LOCATION AND FK_CVIOLATIONACTIVITY_LOCATION)*	HBASIC owner cannot delete HHANDLER row.	TRANS_CODE EQ **D**. If **True**, Match first 2 characters of HANDLER_ID against ACTIVITY_LOCATION in TSIMPLEMENTER_OF_RECORD table where tablename in TSIMPLEMENTER_OF_RECORD EQ HBASIC. Determine if AUTHORIZED_AGENCY EQ owner group for Translator. If **True**, reject record.	
CA008	HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE AND FK_CVIOLATIONFK_HBASICHANDLER*), ACTIVITY_LOCATION (*FK_CENFORCEMENT_LOCATI AND FK_CVIOLATIONACTIVITY_LOCATION*), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)	CLN_COMPLIANCE_SCHEDULE row not found; Cannot delete.	TRANS_CODE EQ **D** AND (HANDLER_ID (*FK_CENFORCEMENTFK_HBASICHANDLE AND FK_CVIOLATIONFK_HBASICHANDLER*), ACTIVITY_LOCATION (*FK_CENFORCEMENT_LOCATI AND FK_CVIOLATIONACTIVITY_LOCATION*), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AGENCY (FK_CENFORCEMENTAGENCY) DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)) **NOT** found in CLN_COMPLIANCE_SCHEDULE . If **True**, reject record.	This checks to ensure there is a row to delete.
CA010	COMPLIANCE_SCHED_DATE	COMPLIANCE_SCHED_DATE must be blank or a valid date.	COMPLIANCE_SCHED_DATE not blank or a valid date. If **True**, reject record.	
CA012	COMPLIANCE_SCHED_DATE	COMPLIANCE_SCHED_DATE must be on or after the ENFORCEMENT_DATE.	COMPLIANCE_SCHED_DATE less than ENFORCEMENT_DATE. If **True**, reject record.	

ERROR #	FIELD NAME(s)	MESSAGE	LOGIC	NOTES
CA040	HANDLER_ID (FK_CENFORCEMENTFK_HBASICHANDLE), ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI), ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT), ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ), AND AGENCY (FK_CENFORCEMENTAGENCY)	Ref. Integrity Viol. (CENFORCEMENT and TR_CENFORCEMENT); Cannot process CLN_COMPLIANCE_SCHEDULE record.	HANDLER_ID (FK_CENFORCEMENTFK_HBASICHANDLE) and ACTIVITY_LOCATION (FK_CENFORCEMENTACTIVITY_LOCATI) and ENFORCEMENT_DATE (FK_CENFORCEMENTENFORCEMENT_DAT) and ENFORCEMENT_SEQ (FK_CENFORCEMENTENFORCEMENT_SEQ) and AGENCY (FK_CENFORCEMENTAGENCY) NOT found in CENFORCEMENT or TR_CENFORCEMENT. If **True,** reject record.	Referential integrity check.
CA042	HANDLER_ID (FK_CVIOLATIONFK_HBASICHANDLER), ACTIVITY_LOCATION (FK_CVIOLATIONACTIVITY_LOCATION), DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) AND VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ)	Ref. Integrity Viol. (CVIOLATION and TR_CVIOLATION); Cannot process CLN_COMPLIANCE_SCHEDULE record.	HANDLER_ID (FK_CVIOLATIONFK_HBASICHANDLER) and ACTIVITY_LOCATION (FK_CVIOLATIONACTIVITY_LOCATION) and DETERMINED_BY_AGENCY (FK_CVIOLATIONDETERMINED_BY_AGE) and VIOLATION_SEQ (FK_CVIOLATIONVIOLATION_SEQ) NOT found in CVIOLATION or TR_CVIOLATION. If **True,** reject record.	Referential integrity check.
–	NOTES	–	–	There are no edits on this data column.